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

На правах рукописи

Дойникова Елена Владимировна

Оценка защищенности и выбор защитных мер в компьютерных сетях на основе графов атак и зависимостей сервисов

Специальность: 05.13.19 – Методы и системы защиты информации, информационная безопасность

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель

д.т.н., профессор

Котенко И.В.

Санкт-Петербург – 2017

Содержание

Введение

Актуальность темы диссертационной работы. В настоящее время, в большинстве коммерческих и государственных организаций, в том числе на промышленных предприятиях (в областях электроэнергетики, машиностроения, систем жизнеобеспечения и т.п.), от которых напрямую зависит безопасность и жизнедеятельность населения, применяются компьютерные сети (КС). Это создает дополнительные возможности для осуществления угроз в отношении таких систем различного рода нарушителями, с целью получения коммерческой выгоды, террористическими или другими целями. Что подтверждается высоким уровнем киберпреступности в России (и в мире). По отчету международной компании по предотвращению и расследованию киберпреступлений Group-IB за 2014 год [5] рынок киберпреступности в России и СНГ в 2011 году составил 2,055 млрд долларов, в 2012 году — 1,938 млрд, и 2,501 млрд в 2013 году.

Хотя в области информационной безопасности (ИБ) предпринимаются серьезные усилия, раскрываемость киберпреступлений по прежнему остается достаточно низкой. Согласно отчету группы компаний по безопасности NTT Group за 2013 год [72], 54% зафиксированных ими инцидентов в компаниях по всему миру, направленных на перехват управления системами, остались не обнаруженными, 71% вредоносного программного обеспечения (ПО), направленного на кражу денег и информации, остался не обнаруженным стандартными средствами, что указывает на неэффективность базовых средств безопасности. Это может быть связано с тем, что предпочтения киберпреступного сообщества сместились в сторону более изощренных целенаправленных атак [32, 82]. Основной целью таких атак являются категории предпринимательства (24%) и финансов (19%) [82].

Необходимость обеспечения ИБ информационных технологий (ИТ) организаций отмечена в стандарте ГОСТ Р ИСО/МЭК ТО 13335-5-2006 [7]. Это обусловлено тем, что «государственные и коммерческие организации в большой степени полагаются на использование информации при ведении своего бизнеса. Нарушение таких характеристик информации и услуг, как конфиденциальность, целостность, доступность, неотказуемость, подотчетность, аутентичность и достоверность, может иметь неблагоприятное воздействие на деловые операции и бизнес организации.

Поэтому существует необходимость в обеспечении безопасности информации и управлении безопасностью систем ИТ в пределах организации.»

Таким образом, повышение защищенности информационных систем (ИС) организаций является важной и актуальной задачей ИБ.

Серьезную проблему с точки зрения обеспечения ИБ представляют открытые распределенные сети с большим количеством узлов. Наличие доступа к системе для большого количества устройств, а также уязвимости ПО компьютеров, объединенных в сеть, создают благоприятную среду для реализации атак. NTTGroup отмечает, что в 2013 году 43% всех зафиксированных атак было направлено против открытых распределенных систем, таких как образовательные платформы, медицинские ИС, электронные услуги и т.п. [72]. Специалисты Symantec также отмечают рост количества веб-атак: в 2011 году в день блокировалось 190370 таких атак, а в 2012 уже 247350 [82].

Одной из причин роста числа реализованных атак является рост количества уязвимостей ПО. По данным компании Symantec, в 2010 году было обнаружено 6253 новых уязвимостей, в 2011 — 4989, а в 2012 — 5291 [82]. 50% уязвимостей, зафиксированных в использовании в 2013 году, были впервые открыты и зафиксированы между 2004 и 2011 годами, что указывает на временной пробел между обнаружением и устранением уязвимостей, который позволяет нарушителям воспользоваться ими [72]. С другой стороны, их легко можно было бы устранить при грамотном управлении ИБ и избежать связанных с ними угроз.

От безопасности и надежности ПО зависит широкий спектр критических приложений и инфраструктур, от систем управления процессами до коммерчески используемых продуктов. Уязвимости данного ПО могут подвергнуть опасности интеллектуальную собственность, доверие потребителей, бизнес-операции и сервисы.

В стандарте ГОСТ Р ИСО/МЭК ТО 13335-3-2007 сказано «если организация не уверена в том, что функционирование ее систем информационных технологий абсолютно не критично к внешним угрозам, она может впоследствии встретиться с серьезными проблемами.» Примером этого может служить атака *Massive Data Exfiltration via SQL Injection* на XYZ National Bank, когда одно незащищенное поле стоило компании примерно 200000 $. Причем базовое сканирование логов и предупреждений помогло бы обнаружить SQL-инъекцию и инцидент обошелся бы в

24980 $. А если бы уязвимость к SQL-инъекции была устранена в процессе разработки, то компания бы не понесла убытков [72].

Поэтому важной задачей обеспечения ИБ организаций является выявление уязвимостей ПО и возможности их использования в компьютерных атаках для нанесения ущерба организации. Для этой цели успешно применяются графы атакующих действий. Однако, приведенный выше пример показывает, что важно не только уметь выявить уязвимости, которые могут использоваться для атаки на КС, но и оценить их возможное влияние на бизнес-операции.

В связи с увеличением количества различных распределенных компьютерных приложений, необходимых для поддержания бизнес-процессов организаций, и развитием концепции сервис-ориентированных архитектур (СОА), не всегда просто определить как именно та или иная уязвимость повлияет на деятельность организации. Ответом на эту проблему стали подходы к определению распространения ущерба в информационной структуре организации на основе графов зависимостей сервисов. Учет распространения ущерба через зависимости сервисов позволит отрегулировать затраты на безопасность, чтобы они не превысили возможный ущерб, не упустить важные уязвимости, которые могут привести к серьезным последствиям, а также обосновать затраты на безопасность. То есть необходимо достичь баланса между затратами на безопасность и возможными потерями в случае успешной реализации атаки. Кроме того, важно не только измерить защищенность, но и предоставить инструменты формирования отчетов и реагирования на инциденты безопасности, что позволит специалистам-практикам быстро и эффективно реагировать на целенаправленные атаки [32].

Хотя большинство организаций осознают важность обеспечения ИБ, реальное финансирование зачастую не соответствует подобным заявлениям. Причем большая часть финансирования обычно уходит на приобретение базовых технических средств защиты, а не на обеспечение процессного подхода к управлению безопасностью. Преимущество процессного подхода состоит в том, что он позволяет учесть эффективность средств защиты и возможные потери в случае успешных компьютерных атак и предоставляет на их основе обоснование затрат на безопасность в виде показателей защищенности. Показатели защищенности являются количественными индикаторами атрибутов безопасности ИС или технологии и предоставляют способ

измерения качества продуктов или сервисов и улучшения процесса. Поэтому подход к управлению безопасностью должен основываться на измерении защищенности в виде надежных, выражаемых количественно показателей. Измерение защищенности является сложной задачей из-за неопределенности входных данных и сложности взаимосвязей внутри защищаемых ИС.

Другой проблемой современных ИС является огромное количество информации и событий безопасности. По данным [72], среднее количество сообщений, генерируемых различными устройствами за день может достигать от нескольких десятков (приложения, прокси серверы, системы контроля доступа, системы обнаружения вторжений и реагирования на вторжения) и сотен (межсетевые экраны, маршрутизаторы, центральные процессоры) тысяч до нескольких миллионов (базы данных). Обработать такое количество информации вручную практически невозможно. Кроме того, в случае, если система подвергается атаке, важным становится временной аспект. Следовательно возникает проблема автоматизированной обработки информации, ее представления в удобном для оператора виде и автоматизированного выбора защитных мер. В рамках решения данной проблемы на текущий момент активно развиваются системы мониторинга безопасности и управления инцидентами (Security Information and Events Management, SIEM). SIEM-системы обычно включают базовые показатели, оценивающие количество уязвимостей, инцидентов и т.п. [1, 44] Такие показатели не дают полной картины информационных и бизнес-рисков, порождаемых потенциальными угрозами, и не позволяют принять всесторонне обоснованное решение по выбору и внедрению защитных мер (контрмер).

Подход к оценке защищенности КС и выбору защитных мер, основанный на комплексной системе показателей и алгоритмов их вычисления с применением графов атак и зависимостей сервисов, которые позволят учитывать различные характеристики компьютерных атак, а также различные аспекты функционирования защищаемой системы и выбирать наиболее эффективные защитные меры, и применимый для SIEM-систем, в целом приведет к существенному повышению эффективности реагирования на инциденты ИБ. Это поможет повысить защищенность программных продуктов и процессов. Таким образом, задача разработки такого подхода является актуальной.

Степень разработанности темы диссертационной работы. Вопросам оценки защищенности КС, анализа рисков и выбора защитных мер посвящено большое

количество государственных стандартов [6–11], документов [4, 28], коммерческих стандартов [129, 134], и работ как отечественных исследователей: С.В. Симонова [40], И.В. Котенко [12–14, 17–27, 29, 31], С.А. Петренко [40], И.Б. Саенко [25, 96], А.М. Астахова [3], Д.С. Черешкина [42], А.Г. Остапенко [16], М.В. Степашкина [17, 24, 27, 37], А.А. Чечулина [12, 20, 23, 43], так и зарубежных: R.P. Lippmann [81, 98], H. Debar [74, 89, 90], N. Kheir [89–91], M. Frigault [70], N. Poolsappasit [125], M. Jahnke [85], G.G. Granadillo [74].

Анализ работ показал, что хотя текущие исследования предлагают для оценки защищенности и выбора контрмер большое количество различных показателей и методик их вычисления, в том числе на основе графов атак и зависимостей сервисов, не существует единого подхода к выбору, определению и вычислению показателей защищенности, позволяющего учесть различные данные при формировании показателей, и оценивать защищенность на различных этапах функционирования системы. Отсюда вытекает первое противоречие — необходимость количественного обоснования затрат на управление безопасностью на основе измерения текущего уровня защищенности системы с целью выбора эффективных защитных мер, и отсутствие четкого подхода к определению и вычислению показателей защищенности.

Кроме того, хотя на данный момент существует большое количество исследований в области выбора защитных мер для реагирования на компьютерные атаки, не существует коммерческого решения в рамках SIEM-систем, использующего все их возможности. Кроме того, предлагаемые методики, как правило, ограничиваются детальным исследованием только одного из наборов характеристик атак и принимаемых защитных мер, например, уровнем навыков атакующего, потенциалом атаки, возможным ущербом и т.п. Таким образом, второе противоречие состоит в необходимости измерения и оценки защищенности на основе адекватных количественных показателей с учетом множества данных, предоставляемых SIEM-системами, и автоматизированного выбора защитных мер на основе комплекса данных показателей, и отсутствии таких показателей и методик в современных SIEM-системах.

Разрешение двух описанных противоречий поможет повысить защищенность программных продуктов и процессов. Поэтому в данной работе предполагается разработать подход к оценке защищенности КС и выбору защитных мер, применимый для SIEM-систем. Предлагаемый подход предполагает разработку комплексной системы

показателей и алгоритмов их вычисления на основе графов атак и зависимостей сервисов. Он подразумевает использование более сложных алгоритмов вычисления показателей при статическом режиме работы, учет информации о событиях безопасности в режиме, близком к реальному времени, а также быстрый перерасчет показателей на основе новой поступающей информации, что позволит отслеживать направление и уровень сложности атаки, ее цели и характеристики атакующего и выбирать наиболее эффективные защитные меры. Такой подход в целом приведет к существенному повышению эффективности реагирования на инциденты ИБ.

Научная задача. Разработка модельно-методического аппарата для оценки защищенности КС и выбора защитных мер на основе совместного применения графов атак и зависимостей сервисов для SIEM-систем.

Объектом исследования в данной работе являются компьютерные сети, атаки на КС с использованием уязвимостей их программного и аппаратного обеспечения.

В качестве **предмета исследования** выступают модели, методики и алгоритмы оценки защищенности КС и выбора защитных мер на этапах проектирования и эксплуатации с использованием показателей защищенности.

Основной **целью** работы является повышение защищенности КС за счет усовершенствования методик, моделей и алгоритмов оценки защищенности КС и выбора контрмер на основе вычисления показателей защищенности. Для достижения цели в исследовании поставлены и решены следующие **задачи**:

1) Анализ показателей защищенности и методик их вычисления на основе моделей умышленных атак в КС в виде графов атакующих действий и моделей распространения воздействия атак в сети в виде графов зависимостей сервисов.

2) Разработка комплекса показателей защищенности с учетом различных входных данных, таких как характеристики КС, атакующих действий, нарушителей и инцидентов безопасности, и на различных уровнях функционирования защищаемой системы (статическом и динамическом).

3) Разработка методики оценки защищенности КС на основе графов атак и зависимостей сервисов.

4) Разработка алгоритмов вычисления показателей защищенности.

5) Разработка методики выбора защитных мер для реагирования на компьютерные атаки с учетом доступных данных.

6) Построение архитектуры и реализация программного прототипа системы оценки защищенности КС и выбора защитных мер на основе предложенной методики.

7) Экспериментальная оценка предложенных алгоритмов и методик, и сравнение их с существующими аналогами.

На защиту выносятся следующие **результаты**:

1) комплекс показателей защищенности компьютерных сетей на основе графов атак и зависимостей сервисов;

2) методика оценки защищенности КС на основе графов атак и зависимостей сервисов;

3) методика выбора защитных мер на основе графов атак и зависимостей сервисов;

4) Архитектура и программная реализация системы оценки защищенности КС и выбора защитных мер на основе предложенных методик.

Используемые методы исследования. В работе используются методы теории множеств, теории вероятностей, теории принятия решений, теории графов, методы оптимизации, а также методы экспертного анализа.

Научная новизна диссертационной работы заключается в следующем:

1) Разработанный комплекс показателей защищенности отличается иерархическим способом классификации на основе объектов оценки, этапов процесса оценки защищенности и категорий показателей (базовые, 0 дня, стоимостные), и позволяет для каждой выделенной группы показателей получить оценку защищенности системы и выбрать защитные меры.

2) Предложенная методика оценки защищенности на основе графов атак и зависимостей сервисов отличается тем, что на каждом уровне иерархии задается совокупность используемых моделей, показателей и алгоритмов оценки, и определяет взаимосвязи между разными уровнями. Методика основана на использовании входных данных о сети и ее уязвимостях, атаках, зависимостях сервисов, атакующих, событиях, контрмерах, экспертных оценках уязвимостей и контрмер, и оценках из открытых баз данных.

3) Разработанная методика выбора защитных мер отличается возможностью генерации комплекса защитных мер на основе доступных входных данных и его последующего уточнения за счет применения иерархического комплекса показателей на

основе анализа событий безопасности, выделением этапов статического и динамического уровня, и совместным применением графов атак и зависимостей сервисов.

4) Разработанная архитектура и программная реализация системы оценки защищенности и выбора защитных мер отличается наличием интерфейсов взаимодействия с SIEM-системами и применением оригинальных методик оценки защищенности и выбора защитных мер.

Теоретическая и практическая ценность диссертационной работы. В настоящее время КС активно развиваются и применяются во многих критически важных коммерческих и государственных отраслях, таких как коммерция и финансы, здравоохранение, образование и др. Так, согласно стратегии развития отрасли ИТ в Российской Федерации (РФ) на 2014 – 2020 годы [38], данная сфера может возрасти в несколько раз в ближайшие десять лет (это касается, в том числе, телефоммуникационной инфраструктуры), и привести к значительному прогрессу в экономике, фундаментальных научных исследованиях, области предоставления государственных услуг (в том числе медицины и образования) и оборонной промышленности. Это создает большие возможности для реализации компьютерных угроз с различными целями, от получения финансовой выгоды, до угроз террористического характера. Поэтому в указе отмечена важность обеспечения ИБ, что невозможно без эффективной оценки текущего уровня защищенности. Кроме того, ввиду активного распространения SIEM-систем, одной из ключевых задач которых является предоставление администратору системы актуальной и адекватной информации о защищенности системы, существует необходимость разработки для них эффективных количественных показателей для формирования текущей картины по безопасности и рекомендаций по реагированию.

Разработанные методики оценки защищенности КС и выбора защитных мер развивают теоретические положения в данной области и позволят снизить уровень возможных потерь в результате компьютерных атак за счет постоянного отслеживания и пересчета показателей защищенности в соответствии с поступающими данными о событиях в системе и своевременного применения адекватных контрмер. Это указывает на обширную область применения результатов данного исследования.

Реализация результатов работы. Исследования, представленные в данной работе, были проведены в рамках следующих научно-исследовательских работ: проекта «Разработка моделей, методик и алгоритмов автоматизированного реагирования на инциденты в процессе управления информацией и событиями безопасности» (грант Российского Фонда Фундаментальных Исследований (РФФИ) № 16-37-00338-мол_а, 2016-2017); проекта «Управление инцидентами и противодействие целевым кибер-физическим атакам в распределенных крупномасштабных критически важных системах с учетом облачных сервисов и сетей Интернета вещей» (грант Российского научного фонда № 15-11-30029, 2015-2017); проекта седьмой рамочной программы (FP7) Европейского Сообщества «Управление информацией и событиями безопасности в инфраструктурах услуг (MASSIF)», контракт № 257475, 2010-2013; проектов Минобрнауки России в рамках Программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы» «Разработка технологий интерактивной визуализации неформализованных данных разнородной структуры для использования в системах поддержки принятия решений при мониторинге и управлении информационной безопасностью информационно-телекоммуникационных систем» (государственный контракт № 14.604.21.0137, 2014-2016) и «Перспективные методы корреляции информации безопасности и управления инцидентами в критически важных инфраструктурах на основе конвергенции технологий обеспечения безопасности на физическом и логическом уровнях» (государственный контракт № 14.616.21.0028, 2014-2014); проекта по программе фундаментальных исследований отделения нанотехнологий и информационных технологий РАН «Архитектурно-программные решения и обеспечение безопасности суперкомпьютерных информационно-вычислительных комплексов новых поколений» «Математические модели, методы и алгоритмы моделирования атак, анализа защищенности компьютерных систем и сетей, анализа рисков безопасности информации и принятия решений о выборе механизмов защиты в компьютерных системах и сетях» (2012-2014); проекта Минобрнауки России «Исследование и разработка методов, моделей и алгоритмов интеллектуализации сервисов защиты информации в критически важных инфраструктурах» (государственный контракт № 11.519.11.4008, 2011-2013); и др.

Полученные результаты использовались в рамках проекта седьмой рамочной программы (FP7) Европейского Сообщества (контракт № 257475), внедрены в учебный процесс СПб ГУТ, используются в научно-инновационной деятельности в ООО «Ароматы безопасности», применяются в рабочем процессе ГК «Омега».

Апробация результатов работы. Основные положения и результаты диссертационной работы докладывались на следующих научных конференциях: международный симпозиум по безопасности мобильного Интернета MobiSec-2016 (Тайчжун, Тайвань, 2016); 24-я международная конференция по параллельной, распределенной и сетевой обработке PDP-2016 (Ираклион, Крит, Греция, 2016); 10-ая международная конференция по рискам и безопасности Интернета и систем CRiSIS-2015 (Митилини, Лесбос, Греция); 22-я международная конференция по параллельной, распределенной и сетевой обработке PDP-2014 (Турин, Италия, 2014); XVI международная заочная научно-практическая конференция (Новосибирск, 2013); 22-я общероссийская научно-техническая конференция «Методы и технические средства обеспечения безопасности информации» МТСОБИ-2013 (Санкт-Петербург, 2013); VIII Санкт-Петербургская межрегиональная конференция «Информационная безопасность регионов России (ИБРР-2013)» (Санкт-Петербург, 2013); 8-я международная конференция по доступности, надежности и безопасности ARES-2013 (Регенсбург, Германия, 2013); 7-я международная конференция по интеллектуальному сбору данных и передовым вычислительным системам IDAACS-2013 (Берлин, Германия, 2013); Часть 5-й Российской мультиконференции по проблемам управления (МКПУ-2012) – конференция «Информационные технологии в управлении (ИТУ-2012)» (Санкт-Петербург, 2012); XIII Санкт-Петербургская Международная Конференция «Региональная информатика-2012 (РИ-2012)» (Санкт-Петербург, 2012); 19-я международная конференция по параллельной, распределенной и сетевой обработке PDP-2011 (Айя-Напа, Кипр, 2011); и др.

Публикации. По материалам диссертационной работы опубликовано более 40 работ, в том числе 9 [12–15, 17, 19–21, 24] — в изданиях из перечня ВАК [33] («Информационно-управляющие системы», «Безопасность информационных технологий», «Проблемы информационной безопасности. Компьютерные системы», «Известия высших учебных заведений. Приборостроение», «Труды СПИИРАН»), 12 —

в изданиях, индексируемых в международных базах Scopus и Web Of Science, и 5 свидетельств о государственной регистрации программ для ЭВМ.

Структура и объем диссертационной работы. Диссертационная работа объемом 163 машинописных страницы, содержит введение, три главы и заключение, список литературы, содержащий 146 наименования, 17 таблиц, 40 рисунков.

Краткое содержание работы. В первой главе определены место и роль оценки защищенности и выбора защитных мер в задаче повышения защищенности КС. Проведен анализ существующих систем оценки защищенности и выбора защитных мер, в том числе основанных на моделировании атак. Рассмотрены используемые методики и алгоритмы вычисления показателей защищенности и выбора защитных мер. Обоснована актуальность цели исследования: оценка защищенности и своевременный выбор защитных мер позволяет снизить потери в результате предумышленных атак, но в настоящее время не существует единого подхода, который позволил бы получать оценку защищенности на разных уровнях, изменять ее с учетом новой поступающей информации и выбирать защитные меры на ее основе в статическом и динамическом режимах. Для достижения поставленной в исследовании цели предложено использование методики, основанной на многоуровневой системе показателей защищенности и алгоритмах их вычисления. Определены требования к системам оценки защищенности и выбора контрмер. Сформулирована постановка задачи исследования.

Во второй главе описывается разработанный комплекс показателей защищенности. Показатели разделены на уровни в соответствии с используемыми для их вычисления входными данными: показатели топологического уровня определяются на основе конфигурации и топологии КС; показатели уровня графа атак позволяют определить вероятность атаки и возможный ущерб с учетом всех путей атаки; на уровне атакующего вводится возможность сформировать профильный граф атак, который включает атаки, которые может реализовать конкретный атакующий; показатели уровня событий позволяют отслеживать развитие атаки и профиль атакующего по событиям безопасности; уровень выбора контрмер включает показатели, характеризующие защитные меры; интегральный уровень включает показатели, вычисляемые на основе показателей предыдущих уровней, характеризующие защищенность системы в целом и позволяющие выбрать защитные меры.

Представлена методика оценки защищенности, разработанная на основе предложенной системы показателей. Методика основана на так называемом anytime-подходе. Каждый из уровней системы показателей позволяет получить оценку ситуации и уточнять ее с появлением новой информации, применяя алгоритмы соответствующего уровня. Для определения уровня защищенности применяются показатели интегрального уровня и как минимум топологического уровня (остальные уровни являются дополнительными). Описаны основные модели и алгоритмы вычисления показателей. Показатели каждого уровня позволяют определить защищенность системы на основе уровня риска и выбрать адекватные контрмеры в соответствии с доступной информацией. Основной особенностью алгоритмов является иерархическая связь между алгоритмами разных групп: выходные данные алгоритмов группы каждого предыдущего уровня используются в качестве входных данных алгоритмов группы следующего уровня. Это позволяет уточнять показатели за счет новых входных данных. Для формирования связей между уровнями модифицированы существующие модели входных данных и алгоритмы вычисления показателей.

Представлена разработанная модель защитной меры и методика выбора защитных мер, работающая в статическом и в динамическом режимах.

В третьей главе приведена архитектура и общее описание программного прототипа, разработанного для проведения экспериментов. Представлены результаты экспериментов и сравнение предложенных методик с существующими аналогами. Результаты проведенных экспериментов с использованием разработанного прототипа показали, что значения таких свойств методик, как своевременность, обоснованность и ресурсопотребление удовлетворяют предъявляемым требованиям, и что новая система превосходит существующие аналоги по функциональности и качеству анализа.

Положения, выносимые на защиту:

1) комплекс показателей защищенности компьютерных сетей на основе графов атак и зависимостей сервисов;

2) методика оценки защищенности компьютерных сетей на основе графов атак и зависимостей сервисов;

3) методика выбора защитных мер на основе графов атак и зависимостей сервисов;

4) архитектура и программная реализация системы оценки защищенности КС и выбора защитных мер на основе предложенных методик.

Глава 1 Современное состояние проблемы оценки защищенности и выбора контрмер в компьютерных сетях

1.1 Место и роль оценки защищенности и выбора контрмер

1.1.1 Российские стандарты информационной безопасности

Задачи оценки защищенности КС являются задачами управления ИБ. Для определения места оценки защищенности КС в процессе их разработки и функционирования, и для определения требований к разрабатываемому подходу, проанализируем основные стандарты РФ в данной области.

В ГОСТ Р ИСО/МЭК 27001-2006 «Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования» [8] представлены требования и основные этапы разработки систем менеджмента информационной безопасности (СМИБ) организаций. Основные этапы разработки СМИБ: определение области и границы действия СМИБ; определение политики СМИБ, в том числе установка критериев оценки рисков; определение подхода к оценке риска в организации; идентификация рисков; анализ и оценка рисков; определение и оценка различных вариантов обработки рисков; выбор целей и мер управления для обработки рисков; утверждение руководством предполагаемых остаточных рисков; получение разрешения руководства на внедрение и эксплуатацию СМИБ; подготовка Положения о применимости СМИБ. В данном исследовании рассматриваются этапы: анализ и оценка рисков; определение и оценка различных вариантов обработки рисков; выбор целей и мер управления для обработки рисков. При этом необходимо учитывать этапы определения подхода к оценке риска в организации и идентификации рисков, так как они предоставляют необходимые входные данные и, следовательно, определяют требования к разрабатываемому подходу. На рисунке 1 рассмотрен состав этапов разработки СМИБ, относящихся к данному исследованию.

Согласно стандарту, определение подхода к оценке риска, включая определение методологии оценки риска, идентификация рисков, анализ и оценка рисков, а также обработка рисков, являются необходимыми этапами внедрения управления безопасностью в организации. На этапе анализа функционирующей СМИБ необходимо способствовать обнаружению событий ИБ и предотвращать инциденты ИБ. Однако данный стандарт не дает непосредственного описания методологий и носит организационный характер.

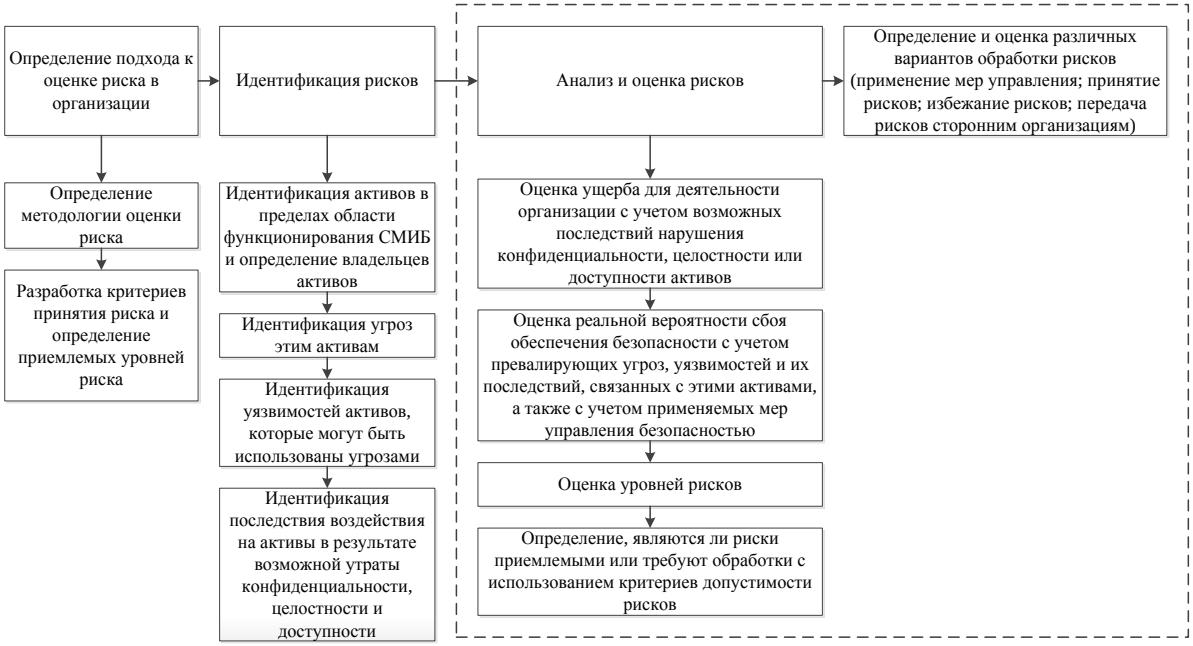


Рисунок 1 – Рассматриваемые в работе этапы управления ИБ

Стандарт «Информационная технология. Методы и средства обеспечения безопасности. Менеджмент информационной безопасности. Измерения» [9] рассматривает вопросы измерения эффективности СМИБ в цикле «Планирование-Осуществление-Проверка-Действие». Полученные измерения должны помочь в усовершенствовании СМИБ и, как следствие, улучшении безопасности организации. В данном исследовании учитываются оценка рисков и выбор контрмер на этапе *планирования*. При этом к целям измерений относятся: верификация степени, до которой были удовлетворены требования безопасности; содействие повышению результативности ИБ с точки зрения общих рисков основной деятельности организации. В дальнейшем, на этапе осуществления, можно измерять эффективность контрмер и пересматривать их состав. Однако это выходит за рамки данной работы.

В ГОСТ Р ИСО/МЭК 13335-1-2006 «Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий» [6] приведены общие понятия и модели управления безопасностью информационных и телекоммуникационных технологий (ИТТ) в организации, описаны основные компоненты безопасности, вовлеченные в процесс управления безопасностью (активы, угрозы, уязвимости, воздействия, защитные меры и риск), и их связи (рисунок 2).

Активы (материальные активы, информация, ПО, способность производить продукт или предоставлять услугу, люди, нематериальные ресурсы) несут ценность для организации. Ценность определяется владельцами активов. Активы могут иметь *уязвимости*, которые

могут снизить ценность активов и влияют на требования к защите активов. Для повышения безопасности активов применяются *защитные меры*.

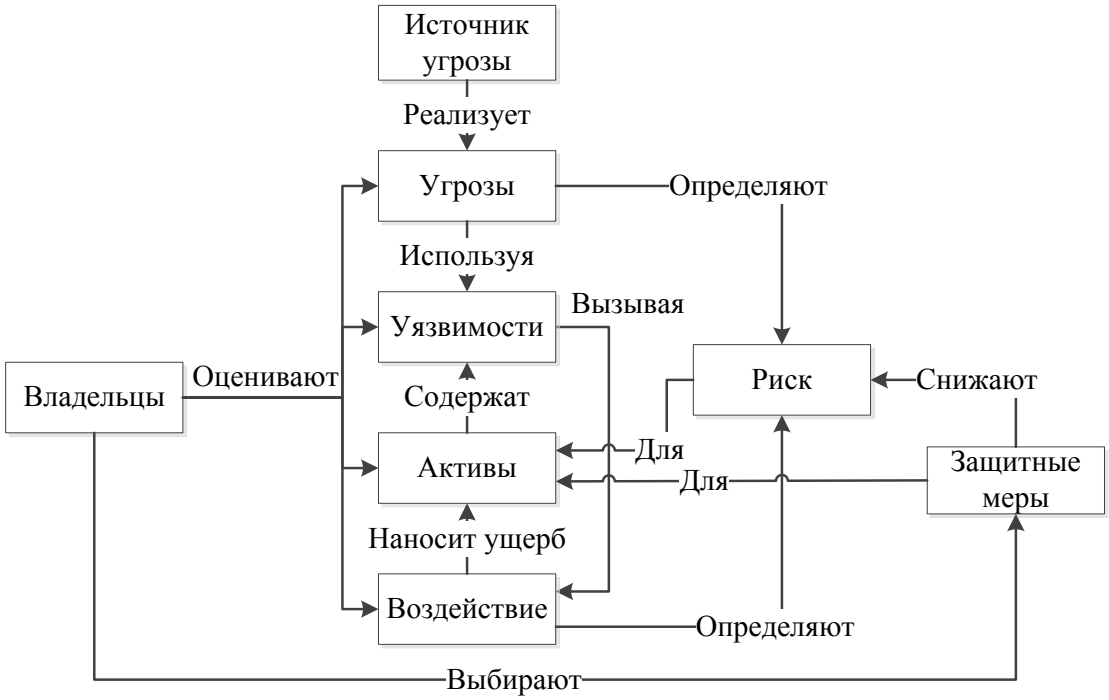


Рисунок 2 – Основные компоненты безопасности и их связи [6]

Угрозы (неавторизованное разрушение, раскрытие, модификация, порча, недоступность или потеря) наносят ущерб активам посредством использования уязвимостей. В зависимости от источника, выделяются угрозы среды и угрозы, обусловленные человеческим фактором (случайные или целенаправленные). В данном исследовании рассматриваются только целенаправленные угрозы, обусловленные человеческим фактором, источником которых является злоумышленник. *Воздействие* — это результат инцидента ИБ, вызванного угрозой и нанесшего ущерб ее активу (разрушение конкретного актива, повреждение ИТТ, нарушение их конфиденциальности, целостности, доступности, неотказуемости, подотчетности, аутентичности и достоверности). Контроль над воздействием позволяет достичь равновесия между предполагаемыми последствиями инцидента и стоимостью защитных мер, при этом следует учитывать вероятность возникновения инцидента. *Риск* определяется как способность конкретной угрозы использовать уязвимость одного или нескольких активов для нанесения ущерба организации. Риск характеризуется комбинацией двух факторов: вероятностью возникновения инцидента и его разрушительным воздействием. На риск может повлиять изменение активов, угроз, уязвимостей или защитных мер, следовательно методология оценки риска должна включать учет таких изменений. Обработка риска включает в себя устранение, снижение, перенос и принятие риска. *Защитные меры*

включают в себя действия, процедуры и механизмы, способные обеспечить безопасность от возникновения угрозы, уменьшить уязвимость, ограничить воздействие инцидента, обнаружить инциденты и облегчить восстановление активов.

Стандарт дает представление, какие компоненты необходимо учесть при разработке методологии оценки рисков, но не предлагает конкретных методологий.

Методы оценки рисков рассматриваются в ГОСТ Р ИСО/МЭК 27005-2010 «Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности» [10]. Стандарт описывает основные виды деятельности, связанные с менеджментом риска ИБ. На рисунке 3 выделены цветом виды деятельности, которые рассматриваются в исследовании, и методы их реализации. На стадии установления контекста происходит определение критериев оценки риска, критериев влияния (ущерба) и критериев принятия риска. В работе выбираются критерии, принятые в мировой практике.

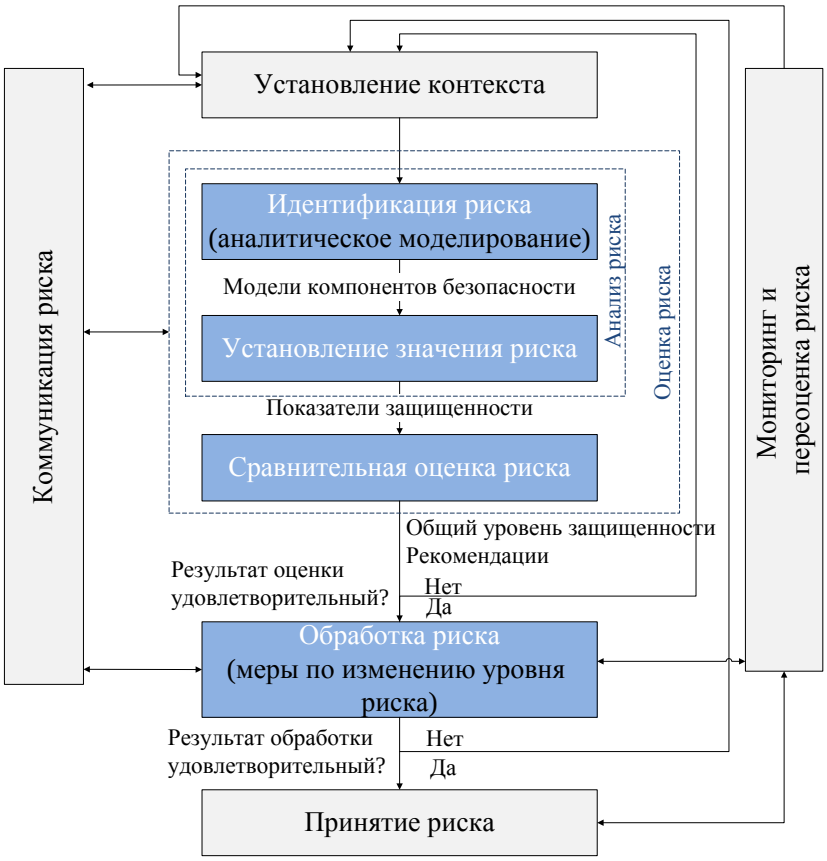


Рисунок 3 – Процесс менеджмента риска ИБ, виды деятельности, рассматриваемые в исследовании, и применяемые методы

В процессе оценки риски должны быть идентифицированы, охарактеризованы количественно или качественно, и для них должны быть назначены приоритеты. Процесс оценки риска позволяет установить ценность активов, выявить потенциальные

угрозы и уязвимости, определить существующие меры и средства контроля и управления и их влияние на риски, определить возможные последствия и назначить приоритеты рискам.

Идентификация риска включает определение активов, определение угроз, определение существующих мер и средств контроля и управления, выявление уязвимостей, определение последствий. Методология установления значения риска может быть качественной, количественной или комбинированной. На этапе обработки риска должны быть выбраны меры и средства контроля и управления риском, на основе стоимости их реализации, их ожидаемой эффективности, и результатов оценки риска.

В стандарте приведены примеры подходов к оценке рисков: высокоуровневая оценка, детальная оценка, комбинированная оценка. Высокоуровневая оценка риска (базовый подход) подразумевает применение стандартных защитных мер ко всем системам ИТ. Преимущество подхода: минимизация времени и средств на выбор защитных мер. Недостаток: применение одинакового базового уровня безопасности для систем ИТ организации, независимо от их критичности.

Детальная оценка риска включает в себя подробную идентификацию и оценку активов, угроз, которым могут подвергнуться эти активы, и оценку уровня их уязвимости. Преимущества подхода: выбор подходящих защитных мер для каждой из систем; подход применим при управлении изменениями в системе обеспечения безопасности. Недостатки: значительные затраты средств и времени; вероятность слишком позднего определения защитных мер для критической системы.

Комбинированный подход подразумевает проведение предварительной оценки риска для выделения систем с высоким уровнем риска. К таким системам применяется детальная оценка риска. Для остальных систем ограничиваются базовым подходом. Преимущество подхода: вложение ресурсов и средств в те области, где это необходимо. Недостаток: возможность ошибочного отнесения системы к тем, которые не требуют детального анализа риска.

Данное исследование относится к организациям с высоким уровнем риска, для которых применяются стратегии детальной оценки рисков. При разработке подхода необходимо учесть недостатки детальной оценки рисков. Предполагается снизить затраты времени и средств за счет создания автоматизированного подхода. Ниже подробнее рассмотрены операции, входящие в детальную оценку.

Для оценки активов стандарт предлагает два способа. Первый подразумевает определение их ценности для организации (например, первоначальная стоимость

актива, стоимость его обновления или воссоздания). При этом важно определить однозначные критерии оценки и шкалу оценки (количественную – например, в денежных единицах, или качественную – например, «Высокая»/«Средняя»/«Низкая»). Другой подход основывается на затратах, понесенных по причине утраты конфиденциальности, целостности или доступности (насколько пострадает деловая деятельность организации и другие активы системы ИТ от утечки, искажения, недоступности и/или разрушения информации). При этом необходимо определить уровни возможного ущерба, связанного с воздействием нежелательного инцидента (например, от нарушения законов, потери репутации, финансовых потерь и т.п.). Также следует учесть зависимости одних активов от других. Выходными данными операции оценки является список активов и их оценок с учетом конфиденциальности, целостности и доступности информации, а также стоимости их замены.

Для оценки угроз необходимо идентифицировать их источники, объекты и оценить вероятность реализации угроз (необходимо учитывать частоту появления угрозы, а также мотивацию, возможности и ресурсы, необходимые потенциальному нарушителю, и степень привлекательности и уязвимости активов системы). В результате формируется перечень идентифицированных угроз, активов, подверженных этим угрозам, и степеней вероятности реализации угроз.

Оценка уязвимостей включает идентификацию уязвимостей, которые могут быть использованы источниками угроз для нанесения ущерба активам, и оценку сложности их эксплуатации.

Идентификация существующих/планируемых защитных мер включает определение их обоснованности, а также совместимости с выбранными после анализа риска мерами.

Величина риска определяется ценностью подвергающихся риску активов, вероятностью реализации угроз, возможностью использования уязвимостей идентифицированными угрозами, а также наличием защитных мер. Метод оценки рисков должен быть повторяемым и прослеживаемым. В стандарте рассматриваются табличные методы оценки риска: матрица с заранее определенными значениями; ранжирование угроз по мерам риска; оценка частоты появления и возможного ущерба, связанного с рисками; разграничение между допустимыми и недопустимыми рисками. Все они состоят в определении уровня риска экспертами на основе качественной шкалы с учетом оценок активов, угроз и уязвимостей. Недостатки таких методов: ручной подход, который может привести к упущению важных деталей; субъективность оценок; качественная шкала оценок, без реальных количественных выражений потерь и рисков.

Выбор защитных мер осуществляется для снижения уровней риска до приемлемых. Необходимо учитывать эффективность и стоимость защитных мер, а также временные ограничения на реализацию. Возможности для снижения уровня риска: избегать риск; уступить риск; снизить уровень угроз; снизить степень уязвимости; снизить возможность воздействия нежелательных событий; отслеживать появление нежелательных событий, реагировать на их появление и устранять их последствия. Защитные меры делятся на организационные и технические. В данном исследовании рассматривается выбор технических мер.

В исследовании необходимо определить конкретные методы реализации операций оценки и обработки риска с учетом следующих требований: подход должен быть применимым для КС; подход должен быть автоматизированным; подход должен быть объективным; шкала оценок должна быть количественной.

ГОСТ Р ИСО/МЭК 31010-2011 «Менеджмент риска. Методы оценки риска» [11] является руководством по менеджменту рисков и методам оценки риска. Он описывает основные этапы оценки рисков и определяет требования к этим этапам. В стандарте представлено подробное описание методов идентификации риска, анализа последствий в процессе анализа риска, анализа вероятностных характеристик и анализа уровня риска, сравнительной оценки риска, и факторов, влияющих на выбор того или иного метода.

В [11] также говорится о необходимости оценки эффективности применяемых методов, анализа чувствительности этих методов и анализа влияния неопределенности исходных данных на итоговую оценку риска.

Выводы по разделу 1.1.1. На основе анализа существующих стандартов в области менеджмента ИБ можно выделить основные этапы оценки риска: установление контекста, оценка риска, обработка риска, принятие риска, коммуникация риска, мониторинг и переоценка риска. В исследовании предлагается сосредоточиться на оценке риска и обработке риска.

Для эффективной обработки риска предпочтительной является детальная количественная оценка риска, которая включает тщательное определение и установление ценности активов, оценку угроз этим активам и оценку уязвимостей. Под риском будем понимать меру ущерба от нежелательного события (в идеале, выраженного в денежных единицах) и возможности того, что это событие произойдет (в идеале, определенную в виде частоты) [140]. Это сложные и трудоемкие процессы, требующие серьезных вложений. Поэтому важной задачей является определение методик количественной оценки и обработки риска, и автоматизация данных процессов, что в исследовании предлагается решить на основе автоматизированного анализа

моделей предметной области, полученных путем аналитического моделирования. В этом направлении был разработан ряд стандартов, которые будут подробно рассмотрены в следующем разделе.

1.1.2 Стандарты в области представления и оценки компонентов безопасности

Для автоматизации идентификации и оценки компонентов безопасности разработаны стандарты унифицированного представления и управления данными по безопасности.

Протокол автоматизации управления данными безопасности SCAP. Национальным институтом стандартов и технологий США (National Institute of Standards and Technology – NIST) разработан протокол автоматизации управления данными безопасности SCAP (Security Content Automation Protocol) [18, 73]. SCAP объединяет ряд стандартов и позволяет составить список используемых в системе платформ и приложений (идентифицировать активы), задать особенности их конфигурации, неблагоприятно влияющие на защищенность, специфицировать список уязвимостей (идентифицировать уязвимости системы), оценить неблагоприятное влияние конфигураций и уязвимостей (определить воздействие), и выявить наиболее критичные уязвимости (оценить уровень риска).

Протокол SCAP версии 1.1 включает:

1) Перечисления, списки или словари, которые задают соглашения по перечислению и именованию, и поддерживаются корпорацией MITRE [111]: «Общие уязвимости и дефекты» (Common Vulnerabilities and Exposures, CVE) – список дефектов ПО; «Общее перечисление платформ» (Common Platform Enumeration, CPE) (версия 2.2) – словарь аппаратного обеспечения, операционных систем (ОС) и приложений; «Общее перечисление конфигураций» (Common Configuration Enumeration, CCE) (версия 5) – словарь конфигураций ПО.

2) Языки спецификации и проверки (определяют способы предоставления инструкций и выражения результатов): «Открытый язык спецификации уязвимостей и оценки» (Open Vulnerability and Assessment Language, OVAL) (версия 5.6) – язык отображения информации о конфигурации системы, оценки состояния и отчета о результатах оценки, используется для определения соответствия политике безопасности, поддерживается MITRE; «Расширяемый формат описания списков контроля

конфигураций» (eXtensible Configuration Checklist Description Format, XCCDF) (версия 1.1.4) – XML-спецификация структурированного набора правил конфигурации, используемых ОС, и платформ, служит для определения безопасных конфигураций, поддерживается Агентством национальной безопасности США (National Security Agency, NSA) и NIST; «Открытый язык отображения проверок безопасности» (Open Checklist Interactive Language, OCIL) (версия 2.0) – используется для определения соответствия работы системы политике безопасности.

3) Стандарт определения показателей для оценки уязвимостей: «Общая система оценки уязвимостей» (Common Vulnerabilities Scoring System, CVSS) (версия 2.0) – задает метод классификации характеристик дефектов ПО и назначения оценок критичности на их основе, поддерживается Форумом групп безопасности и реагирования на инциденты (Forum of Incident Response and Security Teams, FIRST).

В таблице 1 приведены особенности и преимущества протокола. Они позволяют судить о его применимости для автоматизации и стандартизации представления данных для расчета показателей в рамках данного исследования.

Таблица 1 – Особенности и преимущества протокола SCAP

Особенность	Преимущество
Стандартизует то, как компьютеры обмениваются информацией об уязвимостях	Обеспечивает взаимодействие продуктов и сервисов различных производителей
Стандартизует то, какой информацией об уязвимостях обмениваются компьютеры	Обеспечивает возможность использования для продуктов и сервисов различных производителей; снижает зависимость от контекста
Базируется на открытых стандартах	Позволяет использовать совместные решения; применим для большого числа вариантов использования
Использует стандарты управления конфигурациями и активами	Позволяет использовать информацию об активах и конфигурациях для управления уязвимостями
Применим для множества различных методологий управления рисками	Снижает время, усилия и затраты на процесс управления рисками
Содержит детальные ссылки на множество руководящих документов по безопасности	Позволяет автоматизировать процедуры демонстрации соответствия требованиям и подготовки отчетов; снижает возможности недопонимания между руководством, аудиторами и операторами
Основан на стандарте NIST SP 800-53 [127]	Позволяет автоматизировать процедуры демонстрации соответствия руководящим документам, например, Federal Information Security Management Act (FISMA), и формирования отчетности [73]

Так как данное исследование связано с показателями защищенности, рассмотрим подробнее стандарт CVSS.

Общая система оценки уязвимостей CVSS. Стандарт CVSS оценивает уязвимости по шкале со значениями от 0 до 10, используя комбинацию трех комплексных показателей оценки, которые генерируются на основе заданного набора базовых показателей [109].

Показатели базовой оценки *BaseScore* задают обязательные характеристики уязвимости: способ доступа к уязвимости, нужны ли дополнительные условия для ее эксплуатации, и степень потери конфиденциальности, целостности и доступности. Показатели временной оценки *TemporalScore* определяют элементы уязвимости, которые изменяются со временем: доступность кода или методик эксплоита; статус исправления уязвимости; подтверждение технических деталей уязвимости. Показатели контекстной оценки *EnvironmentalScore* описывают результат использования уязвимости в сети определенной организации.

Для получения оценки по CVSS вначале подсчитываются показатели базовой оценки, позволяющие определить фундаментальные характеристики уязвимости. Базовая оценка может быть уточнена значениями показателей временной и контекстной оценки (рисунок 4) [26].

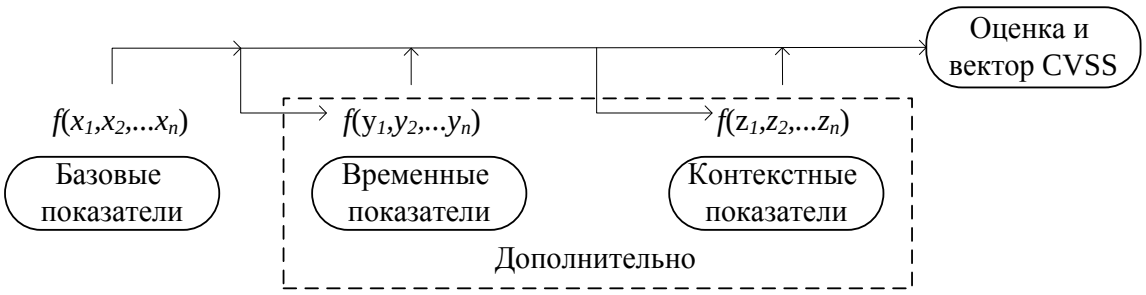


Рисунок 4 – Последовательность оценки CVSS [109]

Базовая, временная и контекстная оценки вычисляется на основе уравнений. Уравнения, а также возможные значения показателей CVSS можно найти в [26, 109].

При определении оценки уязвимостей формируются векторы качественных значений показателей, обеспечивающие открытость оценки по CVSS (раскрывающие, на основе чего она получена). Спецификация каждого вектора состоит из обозначения показателя, за которым следует знак «:» (двоеточие), а за ним – значения показателя. Показатели расположены в определенном порядке и разделены знаком «/» (слэш). Описания базового, временного и контекстного векторов приведены в таблице 2. В квадратных скобках показаны возможные значения каждого показателя.

В 2015 году вышла новая версия CVSS версии 3.0. Однако в существующих базах новая система оценки применяется лишь для вновь добавленных уязвимостей. Поэтому в данном исследовании применяется предыдущая версия CVSS 2.0.

Таблица 2 – Базовый, временной и контекстный векторы CVSS

Вектор	Описание
Базовый	*AV*:[L,A,N]/*AC*:[H,M,L]/*Au*:[M,S,N]/*C*:[N,P,C]/*I*:[N,P,C]/*A*:[N,P,C]
Временной	*E*:[U,POC,F,H,ND]/*RL*:[OF,TF,W,U,ND]/*RC*:[UC,UR,C,ND]
Контекстный	*CDP*:[N,L,LM,MH,H,ND]/*TD*:[N,L,M,H,ND]/*CR*:[L,M,H,ND]/ *IR*:[L,M,H,ND]/*AR*:[L,M,H,ND]

Преимущества CVSS: четкое определение показателей; стандартизованные оценки уязвимостей; CVSS разрабатывается группой профессионалов; открытость системы; возможность приоритезации рисков. Недостатки CVSS: уязвимости оцениваются независимо друг от друга; при оценивании уязвимости учитывается только прямой ущерб для целевого хоста.

Для данного исследования ключевыми факторами являются открытость CVSS оценок, что позволяет использовать их для формирования собственных показателей защищенности, и наличие связей между CVSS и стандартами CPE и CCE, что позволяет автоматизировать идентификацию и оценку уязвимостей.

Тем не менее, необходимо решить проблему корреляции оценок наборов уязвимостей, которые могут использоваться нарушителем для проведения сложных многошаговых атак. Также важным аспектом является учет не только непосредственного ущерба, наносимого уязвимостью, так как он может распространяться через существующие в системе зависимости сервисов.

Другие стандарты оценки уязвимостей и преимущества CVSS. Можно выделить три базовых группы методов оценки уязвимостей [22]:

- Качественное ранжирование – основано на использовании нескольких качественных категорий уязвимостей (например, «низкий», «средний» или «высокий»). Примеры: схема классификации уязвимостей национального центра защиты инфраструктуры США (National Infrastructure Protection Center, NIPC); шкала анализа уязвимостей SANS (SANS Institute's Critical Vulnerability Analysis Scale); система оценки критичности уязвимостей Microsoft Security Bulletin Severity Rating System.

- Количественное ранжирование – основано на использовании количественной шкалы оценки уязвимостей. Примеры: система оценки уязвимостей по стандарту безопасности данных индустрии платежных карт (Payment-Card Industry Data Security Standard, PCI DSS); система оценки уязвимостей US-CERT (United states computer emergency readiness team).

- Применение комплексных показателей оценки уязвимостей. Примеры: система оценки уязвимостей CVSS; система оценки уязвимостей nCircle.

Схема NIPC классифицирует уязвимости по трем категориям рисков: низкий (Low), средний (Medium) и высокий (High). Низкий уровень риска назначается уязвимостям, предоставляющим нарушителю информацию, которую можно использовать в дальнейшем для реализации атак. Средний – уязвимостям, позволяющим локальным или удаленным пользователям повышать привилегии в системе или осуществлять доступ к конфиденциальной информации. Высокий – уязвимостям, позволяющим получить привилегированный доступ к системе [113]. В системе оценки уязвимостей SANS [52] для ранжирования уязвимостей используются понятия критичности (определяется объемом используемых ресурсов и возможностью потерь) и воздействия (определяется степенью влияния наличия уязвимости на безопасность компонентов системы и вероятностью использования дополнительных уязвимостей) уязвимости. Низкий уровень назначается уязвимостям, у которых низкая критичность и низкое воздействие. Высокий – уязвимостям с высокой критичностью и высоким воздействием. Система оценки критичности уязвимостей Microsoft учитывает при назначении оценок сложность использования уязвимости и общее воздействие на безопасность при ее использовании [110].

Основные недостатки вышеперечисленных методов: небольшое число уровней; невозможность агрегирования уровней.

Количественное ранжирование используется для оценки уязвимостей в стандарте PCI DSS [123]. Уязвимости оцениваются по шкале от 1 (уязвимости, которые приводят к возможности раскрытия отдельной информации) до 5 (уязвимости, которые дают возможность удаленным пользователям получить права администратора). Недостаток системы: описания каждого уровня остаются вербальными. Система оценки уязвимостей US-CERT использует значения оценок от 0 до 180 и учитывает такие факторы, как подверженность интернет-инфраструктуры риску и тип предусловий для эксплуатации уязвимостей [137]. Один из основных недостатков системы: относительность оценок уязвимостей, так как они формируются из нечетких значений на основе ответов на заданные вопросы. Система оценки уязвимостей nCircle [114] использует комплексный показатель для оценки уязвимостей, учитывающий количество дней, которые прошли с момента, когда информация об уязвимости впервые стала доступна, угрозу от наличия уязвимости, и набор навыков, необходимый для успешного проведения атаки.

Перечисленным методам в разной степени присущи недостатки: субъективность, неоднозначность, неточность оценивания, нерелевантность контексту использования и отсутствие доверия к оценке бизнес-рисков [114].

Для формирования оценок бизнес-рисков, которым можно доверять, методы оценки уязвимостей должны быть объективными. Методы количественного и качественного ранжирования не удовлетворяют этому требованию. Другими существенными недостатками методов ранжирования уязвимостей являются неоднозначность и неточность получаемых оценок, так как ранги позволяют выявлять только относительное отличие между уязвимостями, и точность ранжирования снижается с ростом количества уязвимостей. Методы оценки на основе показателей в меньшей степени подвержены этим недостаткам, так как показатель представляет атомарное измерение уязвимости, безотносительно других существующих уязвимостей. Тем не менее, оценки, определяемые на основе показателей, остаются субъективными. Методы количественного и качественного ранжирования уязвимостей также подвержены недостатку нерелевантности жизненному циклу уязвимости и контексту применения. В системе nCircle учитывается изменение рисков во времени, но не учитывается контекст применения. Система CVSS не подвержена этому недостатку.

Учитывая перечисленные недостатки и ограничения, а также преимущества системы оценки уязвимостей CVSS, эта система была выбрана как основа для оценки уязвимостей в данном исследовании.

Помимо семейства стандартов SCAP представляют интерес стандарты представления и оценки слабых мест ПО, стандарты представления и оценки атак, стандарты представления и оценки защитных мер.

Стандарт представления слабых мест CWE и стандарт представления атак CAPEC. Стандарт «Общее перечисление слабых мест» (Common Weakness Enumeration, CWE) представляет собой открытый словарь типов слабых мест ПО [62] и включает в себя открытый каталог слабых мест и схему CWE [112]. Словарь позволяет эффективно управлять слабыми местами ПО.

Стандарт «Общее перечисление и классификация шаблонов атак» (Common Attack Pattern Enumeration and Classification, CAPEC) был создан корпорацией MITRE [111] для Министерства национальной безопасности США. Он включает в себя открытый каталог шаблонов атак (версия 2.6) и схему (версия 2.7) [59]. Основным отличием CAPEC является то, что в нем описываются не отдельные уязвимости и слабые места, а подходы и методики, используемые атакующими для компрометации ИС. Подробное описание схемы CAPEC можно найти в [23, 51, 59].

Специалисты MITRE определили ряд возможностей для использования стандарта CAPEC: руководство при определении политик безопасности; руководство при создании требований безопасности; предоставление контекста для анализа рисков;

предоставление контекста для тестирования защищенности; создание связи между разработкой безопасного ПО и безопасными операциями.

Стандарты представления защитных мер CRE и ERI. Протокол SCAP версии 1.2 не предоставляет возможности автоматической реализации защитных мер, хотя идет разработка стандартов «Общее перечисление защитных мер» (Common Remediation Enumeration, CRE) и «Расширенная информация по защитным мерам» (Extended Remediation Information, ERI). Хотя найти базу, использующую данные стандарты, удалось, их можно использовать для создания модели контрмер.

Стандарт CRE является схемой определения и описания защитных мер [71]. Каждая защитная мера описывается с помощью элемента CRE. Элемент CRE представляет собой набор свойств, описывающих защитную меру в формате XML: CRE-ID (идентификатор CRE) – глобальная уникальная строка, связанная с одним элементом CRE (cre:gov.exampleagency:5270-4); текстовое описание элемента, включающее метод и действие защитной меры; список параметров, характеризующих реализацию защитной меры, например, различные виды прав доступа; платформа, представленная с использованием языка CPE, для различных платформ CRE элементы должны отличаться; ссылки на документацию, описывающую защитную меру; метаданные, включающие дату создания элемента, дату изменения элемента, версию и издателя.

Стандарт ERI содержит дополнительную информацию к CRE [86]. ERI включает поля: уникальный идентификатор; ссылка на CRE; индикаторы (ссылки на CCE или CVE); значения, назначаемые параметрам; замены; предусловия; влияние на работу; перезагрузка; метаданные (издатель, действительность).

Выводы по разделу 1.1.2. Стандарты позволяют унифицировать представление основных элементов оценки и обработки риска. Тем не менее, нет единой автоматизированной методики, позволяющей количественно оценить риски и выбрать меры контроля и управления риском в КС с учетом доступной информации о состоянии защищенности КС.

1.1.3 SIEM-система как источник входных данных для оценки защищенности и выбора контрмер

Технологии SIEM-систем активно развиваются. Функции SIEM-систем включают: сбор записей о событиях из различных источников с целью решения задач форензики и

решения проблемы соответствия отчетов нормативно-правовым актам; нормализацию с целью представления записей о событиях из различных источников в едином формате для связи и анализа событий; корреляцию с целью связи записей о событиях и событий от различных систем и приложений для ускорения обнаружения и реагирования на угрозы безопасности; агрегацию с целью снижение объема данных по событиям путем удаления идентичных записей; составление отчетов с целью представления результатов заинтересованным лицам в реальном времени или в форме долгосрочных отчетов.

Путем выполнения данных функций SIEM-системы позволяют повысить эффективность управления безопасностью в организации. Кроме того, SIEM-системы позволяют упростить процесс аудита безопасности за счет решения проблемы согласованности информации по безопасности и процесс обнаружения виновных за счет хранения записей о событиях.

Существует большое количество коммерческих решений в области SIEM-систем. Однако в России рынок SIEM еще недостаточно развит, исключение составляет MaxPatrol SIEM от Positive Technologies [34]. Кроме того, немногие отечественные компании готовы к внедрению подобных систем с точки зрения зрелости систем ИБ и бюджета на ИБ [1].

Хотя SIEM-системы развиваются в сторону анализа и реагирования на инциденты, они не предназначены для детальной оценки рисков [44], этот функционал ложится на сканеры уязвимостей. Существующие SIEM-системы предлагают различные показатели, отражающие текущую ситуацию по защищенности. Эти показатели обычно характеризуют инциденты безопасности и возможные защитные меры на основе сценариев «ЕСЛИ-ТО». А также различные объекты системы с точки зрения безопасности, в том числе количество уязвимостей, инцидентов и т.п. Например, в продукте OSSIM компании AlientVault [100] риск для события безопасности определяется на шкале 0–10 как произведение ценности актива на приоритет события и на надежность события. Такие показатели не дают полной картины информационных и бизнес-рисков, порождаемых потенциальными угрозами, и не позволяют принять всесторонне обоснованное решение по выбору и внедрению защитных мер. Эти задачи ложатся на плечи оператора системы и требуют от него высокого уровня квалификации.

Шаг в сторону расширения возможностей SIEM-систем был сделан в рамках проекта MASSIF [105]. Для повышения эффективности управления защищенностью системы в архитектуру SIEM-системы (рисунок 5) был добавлен компонент оценки защищенности. Он получает скореллированные данные от системы корреляции и обработки и передает их в компонент выбора контрмер. В свою очередь, система

корреляции и обработки получает необработанные данные о событиях через надежную шину данных от компонента сбора данных. Компонет сбора данных получает данные по безопасности в разных форматах с внешних сенсоров. Для представления в реальном времени отчетов о текущем состоянии защищенности, правилах корреляции событий и предложениях от системы выбора контрмер служит компонент визуализации. Решения, предложенные компонентом выбора контрмер и утвержденные при необходимости пользователем через компонент визуализации, отправляются в компонент, ответственный за применение этих контрмер.

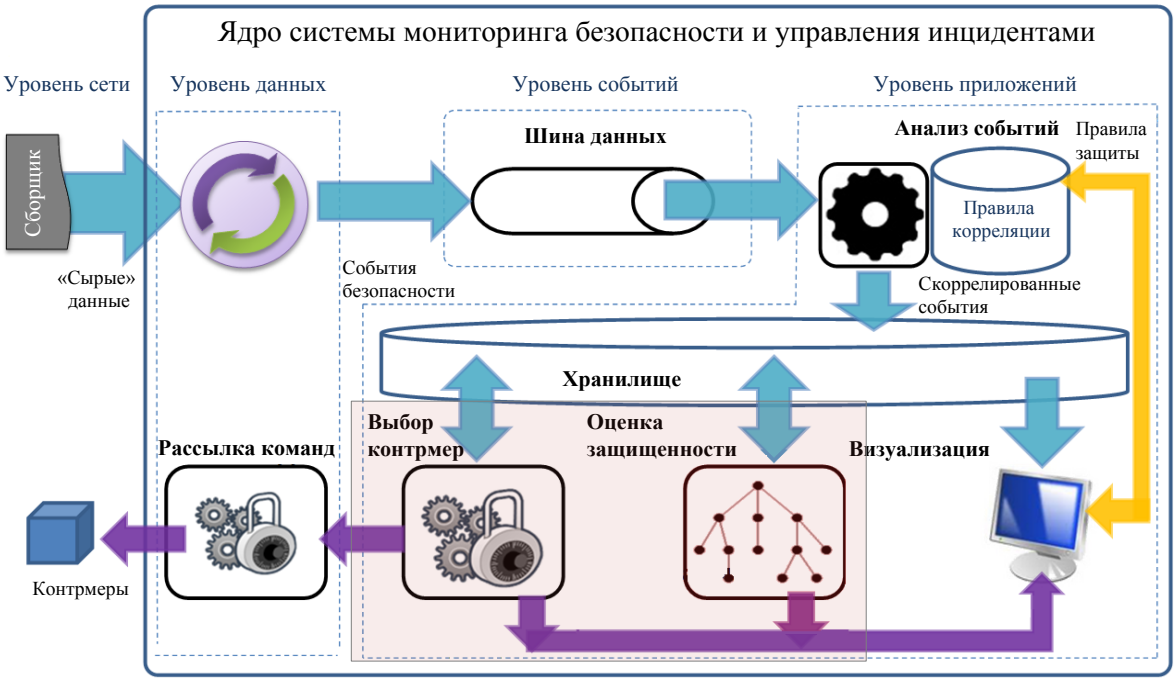


Рисунок 5 – Архитектура SIEM-системы [105]

Выводы по разделу 1.1.3. Современные SIEM-системы предоставляют возможности по эффективному управлению ИБ организаций, однако реализованные в них методики оценки рисков не дают полной картины информационных и бизнес-рисков, порождаемых потенциальными угрозами, и не позволяют принять всесторонне обоснованное решение по выбору и внедрению защитных мер. Поэтому существует необходимость создания новых методик.

1.2 Методики оценки защищенности компьютерных сетей и выбора контрмер

Методики оценки защищенности позволяют представить уровень защищенности системы в форме показателей защищенности. Показатель — мера измерения, дающая

качественную или количественную оценку определенных атрибутов, выведенную на основе аналитической модели, разработанной для определенных информационных потребностей [9]. Количественные методики оценки защищености позволяют измерить риск в терминах денежных единиц и частоты нежелательных событий. На основе полученных измерений можно сравнить стоимость рисков и стоимость реализации защитных мер. Качественные методики ранжируют риски относительно друг друга на основе ценности активов, уязвимостей, угроз и защитных мер.

1.2.1 Качественные методики оценки защищенности

Методика оценки защищенности называется качественной, если в результате ее работы формируется качественная оценка уровня риска. Примеры качественных методик [140]: «Облегченный процесс анализа и оценки рисков» (Facilitated Risk Analysis and Assessment Process, FRAAP) [124]; COBRA [140]; «Оперативная оценка критических угроз, активов и уязвимостей» (Operationally Critical Threat, Asset, and Vulnerability Evaluation, OCTAVE) [56].

Облегченный процесс анализа и оценки рисков FRAAP. Методика FRAAP [124] основана на экспертных знаниях. В процессе FRAAP командой внутренних экспертов под руководством консультанта определяются возможные угрозы целостности, конфиденциальности и доступности информационных ресурсов. Затем устанавливаются приоритеты угроз, на основе вероятностей их успешной реализации за определенный период и возможных последствий для безопасности, в форме качественных значений, например, «Высокий»/«Средний»/«Низкий». Команда полагается на свои знания об угрозах и уязвимостях. Затем команда определяет средства управления, которые могут снизить риски, концентрируясь на наиболее рентабельных. Эти данные документируются и передаются владельцу, принимающему решение о необходимых средствах управления, учитывая природу ресурса, его критичность для бизнес-операций, и стоимость устройства. Результат FRAAP: всеобъемлющий набор документов, который определяет угрозы, их приоритеты по уровням риска, и возможные средства управления для уменьшения уровней риска угроз.

Достоинства FRAAP: использует внутренних экспертов; низкие временные затраты (дни, а не недели или месяцы); эффективен по стоимости; учитывает бизнес-цели владельцев предприятия.

Недостатки FRAAP: команда не пытается получить или разработать показатели защищенности для оценки вероятности угрозы или годовых потерь, за исключением случая, когда данные для их определения легко доступны.

Методика качественной оценки риска COBRA. Методика COBRA была разработана компанией C&A Systems Security Ltd. [54]. Она представляет собой процесс самостоятельного анализа рисков для организаций [140] на основе электронных баз экспертных знаний и процедур логического вывода. Методика совместима с международным стандартом ISO 17799. COBRA включает в себя ПО: Risk Consultant и ISO Compliance. Risk Consultant содержит вопросы, позволяющие определить активы, уязвимости, угрозы и защитные средства организации. ISO Compliance содержит вопросы, позволяющие определить соответствует ли организация стандарту ISO 17799 [84] и рекомендации, которые позволят достичь этого соответствия. Результат работы COBRA: отчеты, которые содержат итоговые оценки информационных рисков и рекомендации по управлению рисками, основанные на общепринятых практиках.

Методика качественной оценки риска OCTAVE. Методика OCTAVE [120] предложена институтом Software Engineering Institute при университете Carnegie Mellon. Она представляет собой набор критериев, которые могут использоваться в качестве основы для методики оценки рисков [140]. Критериями определяется процесс, включающий три фазы: построение профилей угроз на основе активов (с использованием деревьев вариантов); идентификация уязвимостей инфраструктуры (с помощью сканеров или вручную); разработка стратегии безопасности и планов (на основе каталогов защитных мер). Методика основана на использовании экспертных знаний и не требует привлечения сторонних экспертов.

1.2.2 Количественные методики оценки защищенности

Методика количественной оценки риска RiskWatch. Компания RiskWatch разработала методику анализа рисков и семейство программных средств, в которых она реализуется [126]. В качестве критерия для оценки и управления рисками методика использует ожидаемые годовые потери (Annual Loss Expectancy, ALE) и оценку возврата инвестиций (Return on Investment, ROI). RiskWatch ориентирована на точную количественную оценку соотношения потерь от угроз безопасности и затрат на создание системы защиты. В основе продукта RiskWatch находится методика анализа рисков,

состоящая из этапов: (1) определение предмета исследования (защищаемых ресурсов, потерь, угроз, уязвимостей и мер защиты) экспертами путем заполнения таблиц; (2) ввод данных, описывающих конкретные характеристики системы (ценность ресурсов, уязвимости ресурсов, степень уязвимости, частота возникновения угроз, потери и классы инцидентов), вручную или с использованием инструментальных средств; (3) количественная оценка риска (расчет рисков и выбор мер обеспечения безопасности); (4) генерация отчетов.

Достоинства методики: позволяет оценить не только риски, которые существуют у предприятия на данный момент, но и выгоду, которую может принести внедрение механизмов защиты; итоговые отчеты и графики дают материал, достаточный для принятия решений об изменении системы обеспечения безопасности предприятия.

Недостаток: полученный ущерб будет в итоге выше, чем реальный ущерб, т.к. на один и тот же актив может быть направлено несколько угроз, следовательно суммарный ущерб, подсчитанный по угрозам, будет неадекватен реальному, подсчитанному по активам.

Методика оценки риска Microsoft. Компания Microsoft предложила меру оценки уровня защищенности ПО системы — *относительный коэффициент поверхности атаки* (Relative Attack Surface Quotient, RASQ) [138]. Он позволяет квантифицировать относительную подверженность атаке для ИТ активов предприятия. Коэффициент определяется сложением значений эффективной поверхности атаки для всех возможных векторов атаки. Вектора атаки – это характеристики ОС, которые могут положительно или отрицательно влиять на защищенность продукта. Значение эффективной поверхности атаки определяется на основе количества поверхностей атаки внутри вектора атаки (таких как запущенные по умолчанию сетевые сервисы, плохо защищенные учетные записи и файлы и т.п.) и риска компрометации для вектора атаки относительно определенной угрозы. Риск компрометации определяется на основе уязвимости поверхности атаки и ее привлекательности для потенциальных атакующих на шкале от 0 (нет угроз) до 1 (максимальная угроза).

Недостатки точной квантификации рисков: требует чрезмерных временных затрат для определения и верификации или разработки; документация по рискам становится слишком объемной для практики; оценки специфических потерь обычно не требуются для определения, нужно ли управление.

1.2.3 Качественно-количественные методики оценки защищенности

На практике обычно применяется качественно-количественный подход, когда любому качественному уровню соответствуют определенные диапазоны количественных величин [3]. Примеры качественно-количественных методик: CRAMM [64] и методика, предложенная в [27].

Качественно-количественная методика оценки рисков CRAMM. Методика CRAMM [64] была разработана центральным агентством по компьютерам и телекоммуникациям (Central Computer and Telecommunications Agency, CCTA) Великобритании. Методика включает этапы: выявление и оценка активов (это могут быть данные, ПО или аппаратные ресурсы); выявление угроз и уязвимостей и оценка рисков; выявление и приоритезация защитных мер.

Ценность активов определяется в денежных единицах. Для оценки возможного ущерба используется шкала со значениями от 1 до 10. При низкой оценке считается, что рассматриваемая система требует базового уровня защиты и вторая стадия исследования пропускается. Оценка уровней угроз и уязвимостей проводится на основе экспертных знаний с помощью списков вопросов. Выделяются уровни угроз: очень высокий, высокий, средний, низкий и очень низкий. Уровни уязвимости: высокий, средний и низкий. Уровни рисков рассчитываются на основе ожидаемых годовых потерь по шкале от 1 до 7. Ожидаемые годовые потери определяются на основе оценок стоимости активов, уровня угрозы и уровня уязвимости. На основе уровней рисков генерируются варианты защитных мер.

Методика оценки рисков на основе графов атак. В [27] предлагается подход к анализу рисков, состоящий из этапов: построение графа атак; анализ графа атак и оценка защищенности. Методика оценки общего уровня защищенности КС базируется на индексах CVSS и применении некоторых процедур методики анализа рисков FRAAP. Методика включает этапы: (1) вычисление показателей защищенности базовых и составных объектов общего графа атак (критичности хоста, критичности атакующего действия, сложности доступа к уязвимости, степени возможности реализации угрозы); (2) получение качественных оценок уровня риска для всех угроз; (3) оценка уровня защищенности анализируемой КС на основе полученных оценок уровней риска.

Выводы по разделу 1.2. В области оценки защищенности существует большое количество методик, как качественных, так и количественных. Достоинства качественных методик: низкие временные затраты; эффективность по стоимости.

Достоинства количественных методик: возможность количественно оценить и сравнить риски, которые существуют у предприятия на данный момент, и ту выгоду, которую может принести внедрение средств и механизмов защиты; итоговые отчеты и графики дают материал, достаточный для принятия решений об изменении системы обеспечения безопасности предприятия. Недостатки качественных методик: не позволяют получить или разработать показатели защищенности для оценки вероятности угроз или годовых потерь; субъективны, так как основаны на использовании экспертных знаний. Недостатки количественных методик: требуют чрезмерных временных и материальных затрат; документация по рискам становится слишком объемной.

В современных условиях к методикам оценки риска предъявляются дополнительные требования. Количество и сложность атак на ИТ инфраструктуры предприятий растет. При этом даже небольшое время нарушения корректного функционирования ИТ систем может привести к серьезным финансовым потерям, а своевременное обнаружение атакующего и принятие соответствующих защитных мер может их предотвратить. Из-за сложности и размеров современных ИТ инфраструктур, процесс оценки рисков может занять значительное время. Это усложняется тем, что для обоснованного выбора защитных мер, которые позволят минимизировать ущерб от атаки и не нанесут дополнительного вреда, требуются объективные количественные показатели. Таким образом, возникает противоречие: с одной стороны для своевременного и эффективного реагирования на компьютерные атаки необходимо провести полноценную количественную оценку рисков, с другой стороны, для этого требуются серьезные временные и материальные ресурсы. Поэтому существует необходимость в создании автоматизированного подхода, основанного на объективных количественных показателях и позволяющего оценить уровень защищенности системы, выявить ее слабые места и выбрать эффективные защитные меры.

1.3 Показатели защищенности и выбора контрмер и алгоритмы их вычисления

Для представления уровня защищенности системы и выбора защитных мер применяются показатели защищенности. Показатели являются результатом анализа. «Хорошие» показатели должны быть повторяемыми, недорогими при вычислении, численно выраженными, иметь единицы измерения, и соответствовать контексту [129].

В данном исследовании предполагается автоматизировать процесс анализа защищенности и вычисления показателей. Преимущества автоматизации вычисления показателей [129]: точность; повторяемость; надежность; прозрачность. Исследователями были предложены показатели защищенности на основе соответствия критериям, обнаружения вторжений, политик безопасности, инцидентов безопасности, и графического моделирования [121].

1.3.1 Базовые показатели

Под базовыми будем понимать показатели, характеризующие непосредственно элементы конфигурации и безопасности защищаемой системы, такие как система, хосты, программно-аппаратное обеспечение хостов, уязвимости, атакующие действия, угрозы, источники угрозы (атакующие), события безопасности, защитные меры. Для вычисления их значений не требуется построение дополнительных моделей и проведение серьезных вычислений.

В [134] выделяются простые показатели, определяющие количественные, временные и стоимостные характеристики уязвимостей, инцидентов и заплаток. Например: *стоимость инцидентов* (*Cost of Incidents*), *процент систем без известных критичных уязвимостей* (*Percent of Systems with No Known Severe Vulnerabilities*) и другие. Кроме того в [134] выделяются простые показатели, характеризующие конфигурацию системы и приложения системы с точки зрения защищенности: *количество приложений* (*Number of Applications*), *процент критичных приложений* (*Percentage of Critical Applications)* и другие.

В [106] предлагаются показатели для хостов сети: *ценность для бизнеса* (*Business Value*) – показатель, определяемый на шкале от 0 до 100 как ценность самого важного сервиса хоста; *уязвимость хоста* (*Exposure Score*) – оценка возможности того, что хост будет атакован, определяется на шкале от 0 до 1 на основе достижимости хоста и простоты эксплуатации уязвимостей хоста; *оценка риска* (*Risk Score*) определяется на основе показателей *ценность для бизнеса* и *оценка подверженности воздействию* на шкале от 0 до 100; *унаследованный риск* (*Downstream Risk Score*) – суммарный риск для тех хостов, которые можно атаковать с данного хоста.

Простой показатель, характеризующий атакующего, предлагается в [87] – *уровень навыков атакующего* (*Attacker Skill Level, ASL*). Он определяет способность атакующего

выполнить сценарий атаки. Авторы выделяют пять уровней навыков, в соответствии с уровнем атакующего и его/ее знаниях о системе. Уровень атакующего определяется динамически на основе информации о сложности атакующих действий, полученной через инциденты безопасности: $ASL = \max_i \{ASL_i\}$, где ASL_i – сложность i-го атакующего действия.

Простой показатель, характеризующий атаку, предлагается в [46] – ущерб от атаки, который определяется как сумма по всем хостам КС произведений вероятности того, что на хосте есть уязвимость, на ущерб, который будет нанесен в случае компрометации хоста.

Такие показатели позволяют отслеживать изменения, происходящие в системе во времени и иметь представление верхнего уровня о защищенности системы. Однако они не являются достаточными для определения уровня защищенности системы и принятия решений по безопасности, и требуют серьезного дополнительного анализа, проводимого экспертами по безопасности.

1.3.2 Описание графа атак. Показатели на основе графов атак

Графы атак применяются для анализа защищенности системы путем определения того, как могут быть использованы ее уязвимости нарушителями в рамках сложных многошаговых атак. Они отражают все возможные пути атак. Кроме того, они могут отражать состояние системы и переход между состояниями в соответствии с использованными уязвимостями. Для того, чтобы учесть, что уязвимость не всегда может быть использована, вводится показатель вероятности проведения успешного атакующего действия. Он обычно зависит от уровня навыков атакующего и сложности использования уязвимости.

Типы графов атак:

а) Полный граф атак [47] – включает все пути, которыми атакующий может скомпрометировать сеть. Недостаток: сложность $O(n!)$ слишком быстро растет с ростом размера сети.

б) Предиктивный граф [98] – узел добавляется в граф в том случае, если ни один предок данного узла не использует ту же уязвимость для перехода в тоже состояние, что и новый узел. Данные графы строятся намного быстрее, но все еще содержат лишние структуры.

в) Граф со множеством предусловий [81] – включает три типа узлов: состояние; предусловия; уязвимость. Дополнительные циклические дуги добавляются для отображения связей с уже существующими узлами. Преимущество: оперативность построения. Такой граф можно преобразовать в полный или предиктивный граф.

Проблемы при работе с графами атак: высокое время обработки; проблема обработки циклов.

Для решения первой проблемы в ряде работ при построении графов атак предполагается, что они обладают свойством монотонности, что позволяет уменьшить их сложность с экспоненциальной до полиномиальной [81, 129]. Также решение проблемы оперативного построения графов рассматривается в работе [43].

Второй вопрос рассматривается в [66, 125, 141]. В [66] и в [125] предполагается, что для атакующего не имеет смысла повторное посещение уже посещенных узлов. В [141] выделяются виды циклов графа атак и предлагаются методы расчета вероятностей успешности атакующих действий для различных видов циклов.

На основе графов атак был разработан ряд вероятностных моделей для анализа защищенности системы. Для учета неопределенности того, какое именно атакующее действие будет выполнено и насколько успешно в ряде работ предлагается использовать вероятностные графы атак [101, 129, 131, 145], в других работах применяются Байесовские графы атак: [67, 70, 99, 125, 141].

Вероятностный граф атак отражает все возможные пути компрометации системы и показывает распространение вероятности атаки [129]. В [66] предлагается методика количественной оценки защищенности на основе графа привилегий с учетом времени и затрат, необходимых для успешной реализации атаки. Граф привилегий представляет собой набор узлов (каждый узел отображает набор привилегий пользователя) и дуг, отображающих возможность расширения привилегий за счет эксплуатации уязвимостей. Переход из одного состояния в другое происходит при получении новых привилегий. Для оценки времени и затрат применяется Марковская модель. В [131] предлагается показатель уверенности в атаке, который определяет уровень уверенности в том, что атака находится в процессе выполнения, и определяется как отношение количества повторений шаблона атаки к общему количеству атак с соответствующим префиксом на графе атак. В [145] для определения уровня уверенности в том, что узел графа атак был достигнут, используется индекс уверенности в компрометации, который определяется на основе значения доверия детектора, сгенерировавшего предупреждение, соответствующее узлу графа, и индексах уверенности в компрометации для прямых потомков: индекс уверенности в компрометации равен значению доверия детектора,

если у узла нет потомков; если потомки соединены связью типа «ИЛИ» индекс уверенности в компрометации равен максимальному индексу для потомков; если потомки соединены связью типа «И» индекс уверенности в компрометации равен минимальному индексу для потомков.

В [101] предлагается иерархический метод количественной оценки риска на основе графа атак. Авторы выделяют несколько уровней сети: уровень уязвимостей, уровень сервисов, аппаратный уровень и сетевой уровень. Уровень риска вычисляется, для уязвимостей, сервисов, хостов и сети, с учетом вероятности атаки. На уровне уязвимостей вычисляются вероятности успешного использования уязвимостей, соответствующих узлам графа атак. Вероятность того, что на j-м шаге атакующий выберет уязвимость v_j, вычисляется по формуле: $\lambda_{ij} = Co_{ij} / \sum_{k=1}^{n} Co_k$, где Co_{ij} – сложность атаки с использованием уязвимости v_j, $\sum_{k=1}^{n} Co_k$ – сумма сложностей атак с использованием всех остальных доступных уязвимостей. Вероятность успешной атаки с узла n_i на целевой узел графа атак n_j за m шагов рассчитывается итеративно по формуле: $p_i^m = \sum_{n_j \in Q(n_i)} \lambda_{ij} \times Co_j \times p_j^{m-1}$, где $Q(n_i)$ – множество всех узлов, доступных с n_i, процесс заканчивается, когда $p_j^{m-1} = 1$.

В [129] рассматривается методика анализа рисков на основе вероятностного графа атак, позволяющая определить наиболее вероятные пути атаки в сети. Авторы используют граф, предложенный в [122], который отражает связи между конфигурацией системы (предусловиями атакующих действий), атакующими действиями и последствиями атакующих действий. В работе предлагается два типа показателей: индивидуальные показатели (характеризуют свойства отдельных компонентов графа, не учитывают отношения между компонентами) и интегральные показатели (учитывают отношения между компонентами). Для вычисления индивидуальных показателей применяются индексы CVSS, полученные из открытой базы уязвимостей NVD [119]. Индивидуальный показатель: условная вероятность успешной атаки в случае, если соблюдены все предусловия, вычисляемая на основе индекса CVSS «вектор доступа»). Интегральный показатель: абсолютная вероятность успешной атаки, вычисляемая на основе перемножения индивидуальных показателей. Для выбора защитных мер интегральные показатели пересчитываются с учетом введения отдельных защитных мер.

Выбираются меры, которые позволяют снизить интегральный показатель вероятности атаки на наиболее ценные хосты.

Данные работы интересны тем, что они используют разные аспекты для определения вероятности атаки (в том числе, время, затраты, информацию от детекторов о событиях, предыдущие шаги атаки, оценки уязвимостей). Но они не останавливаются подробно на других важных аспектах оценки риска (проблемы с циклами на графе, ущерб, наносимый атакующим действием, или автоматизация процесса выбора защитных мер).

Байесовский граф – это граф, узлам которого соответствуют случайные переменные Бернулли, отображающие состояние узла (скомпрометирован или нет). В [141] предлагается показатель вероятности многошаговых атак, использующих комбинации различных уязвимостей. Граф атак представляет собой множество вершин (отображающих действия или условия, необходимые для перехода к следующему действию) и дуг. В работе дается физическая интерпретация показателей вероятности атаки; предлагается применение экспертных оценок уязвимостей (например, CVSS) для определения вероятности; в формуле вычисления полной вероятности учитываются конъюнктивные и дизъюнктивные зависимости на графе; вводится ограничение на независимость событий осуществления действий атакующим.

Каждому узлу, отображающему действие e, соответствует численное значение $p(e)$ – относительная вероятность того, что соответствующее действие будет выполнено нарушителями, когда все требуемые условия соблюдены (интерпретируется как доля нарушителей, которые могут и исполнят e, среди всех нарушителей, которые попытаются скомпрометировать заданную сеть за заданное время). Каждому узлу, отображающему условие c, соответствует численное значение $p(c)$, равное 1. На основе относительных вероятностей рассчитываются $P(e)$ и $P(c)$ – полные вероятности того, что атакующий может успешно достичь узла и выполнить действие или удовлетворить условие на заданном графе атак, соответственно (доля нарушителей, которые будут успешно использовать e или удовлетворять c, соответственно). $P(e) = p(e) \cdot \prod_{c \in R_r(e)} P(c)$, где $R_r(e)$ – множество условий, которые необходимо удовлетворить для успешного осуществления действия e. $P(c)=p(c)$, если $R_i(e) = \phi$; $P(c) = p(c) \cdot \otimes_{e \in R_i(e)} P(e)$, иначе, где $R_i(e)$ – множество действий, которые удовлетворяют условию c; $\otimes$ – оператор, который рекурсивно определяется как $\otimes P(e) = P(e)$ для любого $e \in E$, где E – множество узлов

графа, соответствующих действиям, и $\otimes(S_1 \cup S_2) = \oplus S_1 + \oplus S_2 - \oplus S_1 \cdot \oplus S_2$ для любых дизъюнктных и непустых наборов $S_1 \subseteq E$ и $S_2 \subseteq E$.

В [67] графы атак применяются для вычисления уровня уязвимости критических ресурсов (узлы соответствуют действиям, дуги – связям между ними). Исходные вероятности компрометации узлов графа вычисляются с учетом профиля атакующего (включая уровень навыков, позицию и время). Итоговые апостериорные вероятности компрометации вычисляются на основе Байесовского вывода с учетом исходных вероятностей компрометации. В [121] рассматривается вероятностный подход на основе исторических данных, реализованный на основе байесовского вывода. Авторы рассматривают вычисление таких показателей, как *уровень навыков атакующего* и *вероятность атаки*. В [57] рассматривается автоматическая генерация байесовского графа атак и оценка защищенности сети на основе анализа деревьев недочетов. Авторы предлагают показатель, который позволяет выявить наиболее критичные уязвимости для определения точек внедрения контрмер: *наиболее критичный компонент системы* (*The Most Critical System Component*). Он определяется на основе значения критичности:

$$C_k = P(TE = 1) - P(TE = 1 \mid \pi(v_k) = 0, 0) \; (1 \le k \le n),$$

где v_k – уязвимость системы, $v_k \in V$, V– набор всех уязвимостей системы;

$\pi(v_k)$ – вероятность успешной эксплуатации уязвимости v_k;

$P(TE = 1) = p(s_s = 1) = \sum_{s_1,\ldots s_k} P(s_1,\ldots,s_k, s_s = 1)$ – показатель *ненадежности конечного события*, где s_s – успешное состояние (для атакующего); $s_1,\ldots,s_k$ – набор начальных состояний, $1 \le k < s$.

Уязвимость v_k считается наиболее критичным компонентом тогда и только тогда, когда не существует такой уязвимости v_j ($1 \le j < n$, $j \ne k$), что $C_j > C_k$.

В [125] байесовский граф атак используется для переопределения вероятности атаки на основе поступающих событий. Для определения локальных априорных вероятностей осуществления угрозы S_i с применением уязвимости e_i используются индексы CVSS: $Pr(S_i) = 2 \times B\_AV \times B\_AC \times B\_AU$, где B_AV – вектор доступа CVSS, B_AC – сложность доступа CVSS, B_AU – аутентификация CVSS. Для определения вероятности компрометации узла S_j, учитывая различные комбинации состояний его предков $Pa[S_j]$, используются формулы: $Pr(S_j \mid Pa\left[S_j\right]) =$

$$= \begin{cases} 0, & \exists S_i \in Pa[S_j] \mid S_i = 0 \\ Pr(\cap_{S_i=1} e_i), & \text{иначе} \end{cases},$$ если узлы графа связаны отношением AND;

$Pr(S_j \mid Pa\left[S_j\right]) = \begin{cases} 0, & \forall S_i \in Pa[S_j] \mid S_i = 0 \\ Pr(\cup_{S_i=1} e_i), & \text{иначе} \end{cases}$, если отношением OR. Где $Pr(\cap_{S_i=1} e_i) = \prod_{S_i=1} e_i$, и $Pr(\cup_{S_i=1} e_i) = 1 - \prod_{S_i=1} [1 - Pr(e_i)]$ [99]. Для определения полной вероятности компрометации $Pr(S_j)$ с учетом всех возможных состояний предков S_j $S=\{S_1,...,S_n\}$ используется формула совместного распределения вероятностей: $Pr(S_1,...,S_n) = \prod_{i=1}^{n} Pr(S_i \mid Pa[S_i])$. В случае поступления событий безопасности (оценка в динамическом режиме), апостериорные вероятности рассчитываются на основе теоремы Байеса.

Преимущества применения байесовских графов перед вероятностными: возможность делать выводы об атаке на основе субъективных знаний при отсутствии статистических данных об успешном использовании уязвимостей сети; возможность оценки риска в динамическом режиме на основе поступающих событий.

В данном исследовании за основу предлагается взять подход на основе байесовских графов атак. Рассмотренные работы учитывают различные параметры при определении вероятности атак, однако комплексного решения, позволяющего учитывать различные характеристики атаки в статическом и динамическом режимах работы системы, найти не удалось. Кроме того, в рассмотренных работах не учитываются зависимости между сервисами системы при определении ущерба от атак, финансовые потери в случае успешной реализации атак, затраты на внедрение контрмер. Эти проблемы предполагается решить в исследовании. В ряде работ учитываются временные и исторические аспекты при вычислении вероятности атаки, однако их анализ требует наличия большого количества исходных данных о реальных атаках на КС и выходит за пределы исследования.

1.3.3 Описание графа зависимостей сервисов. Показатели на основе графов зависимостей сервисов

С увеличением роли ИТ в работе организаций, появилась необходимость интеграции бизнес-функций организации и ИТ. Это привело к развитию концепции СОА [35]. В СОА процессы реализуются с помощью сервисов. Сервис – это ресурс, предоставляющий возможность выполнения задач, формирующих необходимую

функциональность с точки зрения поставщиков и потребителей услуг [142]. То есть сервисы ориентированы на бизнес. Зависимости между сервисами позволяют определить обоснованность вложений в ИТ за счет определения связи между бизнес-целями и ИТ. Из-за сложности взаимосвязей между сервисами в СОА не всегда просто определить как та или иная уязвимость повлияет на деятельность организации. Ответом на эту проблему стали подходы к определению распространения ущерба в информационной структуре организации на основе графов зависимостей сервисов [50, 85, 90, 136]. Граф зависимостей сервисов представляет собой множество сервисов КС связанных между собой в соответствии с тем, как свойства безопасности одного сервиса зависят от свойств безопасности другого. Учет распространения ущерба через зависимости сервисов позволит отрегулировать затраты на безопасность, чтобы они не превысили возможный ущерб, не упустить важные уязвимости, которые могут привести к серьезным последствиям, а также обосновать затраты на безопасность.

Показатели на основе графов зависимостей сервисов. В [136] для определения побочного ущерба при реагировании предлагается показатель, отражающий стоимость снижения производительности сервиса в результате потери его доступности. Итоговая производительность сервиса определяется на основе дерева зависимостей между ресурсами.

В [50] показатель ущерба от атаки рассчитывается на основе карты системы (граф зависимостей, объединяющий приоритетные ресурсы), иерархии ресурсов (группировка ресурсов по типам с выделением контрмер для каждого типа) и стоимостной модели (в которой ресурсам назначаются стоимости) как сумма стоимостей узлов системной карты, на которые атакующее действие повлияло негативно.

В [85] получила дальнейшее развитие концепция производительности ресурсов, предложенная в [136]. Стоимость ущерба, нанесенного в результате проведения атаки, рассчитывается как значение доступности ресурсов после проведения атаки (в процентном отношении). Доступность ресурсов зависит от внутренней доступности ресурса и доступности ресурсов, необходимых для его функционирования.

Подход к определению уровня распространяемого ущерба, предлагаемый в [90], включает пять шагов: (1) вычисление ущерба, наносимого отдельному свойству безопасности ресурса (конфиденциальности, целостности или доступности) зависимыми ресурсами, на основе веса зависимости, внутреннего ущерба ресурса и влияния некорректной работы зависимых ресурсов; (2) вычисление совместного распространения ущерба для трех свойств безопасности ресурса (вектора ущерба) на основе матрицы зависимости свойств безопасности ресурса от свойств безопасности

зависимого ресурса; (3) вычисление итогового распространения ущерба для ресурса с учетом конъюнктивных и дизъюнктивных зависимостей графа; (4) определение итогового вектора ущерба ресурса на основе векторов внутреннего ущерба ресурса и влияния некорректной работы зависимых ресурсов; (5) определение непрямого распространения ущерба по графу зависимостей сервисов на основе алгоритма Форда-Фалкерсона. Работа алгоритма начинается с узлов, для которых изменилось значение внутреннего ущерба. Для определения ущерба, нанесенного определенному узлу, значение ущерба домножается на вектор-столбец критичности. В случае, если ущерб нанесен нескольким ресурсам, итоговый ущерб определяется суммой ущерба по всем ресурсам.

Рассмотренные работы показали, что при определении уровня защищенности системы важно учитывать распространение ущерба через зависимости сервисов. Для этого целесообразно применять графы зависимостей сервисов и определяемый на их основе показатель распространенного ущерба. За основу предлагается взять подход, описанный в [90], так как он позволяет учесть распространение ущерба для всех трех свойств безопасности. Однако этот подход не учитывает вероятность нанесения ущерба ресурсу. Эту проблему планируется решить за счет формирования связи между моделями атак и зависимостей сервисов.

1.3.4 Методики и показатели выбора защитных мер

Вопросы выбора защитных мер рассматриваются в [50, 53, 57, 67, 74, 75, 81, 85, 89, 98, 118, 125, 132, 136, 145].

В [53, 75] предложены методики на основе теории игр. Преимущество таких методик: возможность учета временного аспекта. В [53] игра (стратегия) описывается как последовательность состояний и действий. Для описания состояния используется граф зависимостей между сервисами и файлами системы и ряд свойств каждого узла (уязвимость, скомпрометированность и др.). Изменение состояния происходит при применении действий (правил). Для выбора стратегии, минимальной по стоимости и времени, авторы минимизируют сумму затрат и времени, необходимых для выполнения всех действий стратегии. Результатом работы подхода является оптимальная стратегия — наилучшая последовательность действий, которую администратор или нарушитель может применить для достижения цели. В [75] авторы вводят модель атаки-

защиты, основанную на теории игр (Game Theoretical Attack-Defense Model, GTADM) и иерархическую модель вычисления рисков (Hierarchical Risk Computing Model, HRCM). GTADM описывается как некооперативная статическая игра с ненулевой суммой и полной информацией, и применяется для вычисления вероятностей угроз и ожидаемого ущерба в результате успешной реализации каждой угрозы. Игроками являются атакующий и защитник. Защитник стремится найти оптимальную по стоимости стратегию.

В [81, 98, 118] рассматриваются методики принятия решений на основе логического вывода на графах атак. В [118] рассматривается подход к минимизации затрат на повышение уровня защищенности сетей на основе графов зависимостей между эксплоитами. Авторы преобразуют путь на графе до цели атаки в выражение, включающее начальные предусловия атаки в конъюнктивной нормальной форме. Для реализации защитных мер выбираются дизъюнкции, включающие максимальное количество переменных, соответствующих начальным предусловиям атаки. Выбирается набор защитных мер с минимальной стоимостью. Недостаток: при выборе защитных мер учитываются только начальные узлы. В [81] предлагается подход к генерации графа атак с множеством предусловий, позволяющий выявить наиболее критичные уязвимости системы (дающие нарушителю наибольший доступ в системе). Достижимость хоста для нарушителя определяется наличием администраторского или гостевого доступа. Авторы формируют рекомендации по реализации защитных мер на основе показателей, определяющих как много путей атаки будет заблокировано введением защитной меры. Недостаток: в работе не описаны конкретные показатели и не предложено методик их вычисления и методик определения уровня защищенности.

В [98] рассматривается стратегия защиты в глубину на основе применения защитных мер по уровням графа атак. Для оценки защищенности сети предлагается показатель *процент компрометации сети* (*Network Compromise Percentage*, *NCP*), который определяет процент хостов сети, на которых атакующий может получить права пользователя или администратора. Показатель принимает значения от 0 до 100 %. Для выбора защитных мер предлагается алгоритм: граф обходится в ширину и для каждого хоста создается список входящих дуг (соответствующих уязвимостям, которые могут быть использованы для компрометации хоста). Для этих дуг подбираются защитные меры. Кроме того, создаются группы хостов, для которых могут использоваться одни и те же меры: если множество потомков рассматриваемого узла пересекается со множеством потомков узла одной из групп, узел добавляется в данную группу (так как компрометация данного узла усиливает вероятность компрометации его потомков, а

устранение уязвимостей, ведущих к компрометации, напротив эту вероятность уменьшает). Каждая защитная мера устраняет дугу графа. Таким образом, для каждой защитной меры рассчитывается *NCP* после реализации меры и вычисляется разница между предыдущим и новым значением *NCP*. Затем меры упорядочиваются в соответствии с максимальной разницей. Недостаток: в работе не рассматриваются подробные алгоритмы вычисления уровня риска с учетом ущерба и вероятности атаки.

В [67, 125] предлагается выбор защитных мер в динамическом режиме на основе Байесовских графов атак. В [57, 92] описываются методики определения общего уровня защищенности системы, затем используемого для сравнения уровня защищенности до и после внедрения защитных мер.

В [50, 74, 85, 89, 132, 136, 145] рассматривается определение ожидаемых потерь и эффективности защитных мер. В [50] предлагается подход для автоматического выбора контрмер на основе показателей *ущерба от атаки* и *эффективности реагирования*. Для этого используется карта системы (граф зависимостей, объединяющий приоритетные ресурсы), иерархия ресурсов (группировка ресурсов по типам с выделением контрмер для каждого типа) и стоимостная модель (в которой ресурсам назначаются стоимости). Ущерб от атаки вычисляется как сумма стоимостей узлов системной карты, на которые атакующее действие повлияло негативно. Эффективность реагирования определяется как сумма стоимостей узлов, восстановленных в результате реализации контрмеры. Побочный ущерб при реагировании определяется как стоимость узлов, на которые реализация контрмеры повлияла отрицательно. Далее выбирается набор контрмер, которые дадут наибольший выигрыш при наименьших затратах. Авторы предполагают, что им известна полная картина вторжения, и оптимальная цепочка контрмер будет состоять из набора оптимальных контрмер для каждого узла.

В [145] для определения эффективности реагирования применяется показатель жизнеспособности системы, зависящий от того, какое количество операций и целей безопасности системы может быть сохранено в случае вторжения. Вначале выбираются узлы реализации защитных мер. Для выбранных узлов выбираются методы реагирования на основе показателя *индекс реагирования* (*Response Index, RI*): $RI=a \cdot EI - b \cdot DI$, где a и b — параметры развертывания мер, EI — индекс эффективности защитной меры, DI — индекс разрушений, наносимых защитной мерой системе. Выбирается защитная мера с максимальным RI.

В [85] авторы предлагают показатели для анализа затрат на вторжение и реакции на вторжение на основе графа зависимостей между ресурсами: *ущерб от атаки* (рассчитывается как значение доступности после проведения атаки), *затраты на*

реакцию (нормализованное количество ресурсов, которые нужно изменить для реализации реакции), *эффективность реакции* (разница между ущербом от атаки и от реакции) и *дополнительный ущерб от реакции* (рассчитывается также, как ущерб от атаки, но на основе графа доступности, построенного после реализации реакции).

В [89] предлагается показатель выбора контрмер для реагирования на атаки на основе графов зависимостей сервисов – *показатель возврата инвестиций в реагирование* (*Return-On-Response-Investment, RORI*): $RORI = \frac{RG - (CD + OC)}{CD + OC}$, где RG – эффективность реагирования, $RG=IC_b-IC_a$; IC_b – ожидаемые негативные последствия атаки, в случае отсутствия контрмер; IC_a – ожидаемые негативные последствия атаки, в случае реализации контрмер; CD – побочные потери при реагировании; OC – затраты на контрмеры. Для реализации выбирается контрмера с наибольшим значением показателя *RORI*. Преимущество: учет распространения ущерба через зависимости сервисов. В [74] рассмотрены недостатки показателя *RORI* (не учитывается вариант отсутствия контрмер и размер инфраструктуры системы) и предложен улучшенный показатель *RORI*.

В [65, 77, 143] используются экономические индексы для оценки возможных потерь и эффективности контрмер. Преимущество таких методов: установление прямой зависимости между финансовыми затратами организации и потерями от атак. Так, в [77] контрмеры оцениваются по параметрам: выигрыш от реализации контрмеры; затраты на реализацию контрмеры; дополнительная польза от реализации контрмеры. На их основе определяется общий выигрыш от реализации контрмеры.

Существующие работы рассматривают выбор контрмер с помощью различных показателей. Тем не менее не удалось найти методики, которая бы позволяла учитывать различные показатели для выбора защитных мер в зависимости от имеющихся входных данных. Данную проблему предполагается решить в проводимом исследовании за счет развития подходов, применяемых в [98] (выбор наиболее незащищенных точек графа атак), [89] и [77] (учет стоимостных аспектов при выборе контрмер).

1.3.5 Интегральные показатели

Под интегральными будем понимать показатели, дающие общее представление о защищенности системы, такие как *уровень риска* (*Risk Level*) и *поверхность атаки* (*Attack Surface*).

Уровень риска зависит от вероятности успешной реализации атаки (которая, в свою очередь, зависит от уязвимости объекта атаки и вероятности того, что атака будет предпринята) и ущерба, наносимого в результате успешной реализации атаки [10].

В [57, 89, 92, 101, 118, 125, 145] рассматривается аспект оценки уровня риска на основе графов атак и зависимостей сервисов. Работа [101] интересна тем, что в ней выделяется несколько уровней оценки: уровень уязвимостей, уровень сервисов, аппаратный уровень и сетевой уровень. Вначале определяется уровень риска сервисов на основе критичности сервиса для сети и влияния уязвимостей на сервис. На аппаратном уровне определяется коэффициент риска хостов с учетом риска для сервисов хоста, критичности хоста, и влияния уязвимостей на хост. Уровень риска сети *NR* определяется на сетевом уровне как сумма риска для всех хостов. Выводы о защищенности сети делаются на основе степени распределения сетевого риска $NRD = (NR - NR_{\min}) / (NR_{\max} - NR_{\min})$, $0 \le NRD \le 1$, где NR_{max} – максимальное значение риска в сети: $NR_{\max} = \sum_{k=1}^{n} R_{i_k} HI_{j_k}$, где $R = \{R_{i_1}, R_{i_2}, ..., R_{k_n}\}$ – возрастающий набор значений риска без учета степени важности хостов, $HI = \{HI_{j_1}, HI_{k_2}, ..., HI_{i_n}\}$ – возрастающий набор степеней важности хостов; NR_{min} – минимальное значение риска в сети: $NR_{\min} = \sum_{k=1}^{n} R_{i_k} HI_{j_{n-k+1}}$.

В [67, 125] уровень риска рассчитывается с учетом только вероятности компрометации критических ресурсов. В [89] – с учетом только возможного ущерба.

Концепция показателя *поверхность атаки* была развита в работах [79, 102–104]. Поверхность атаки определяется на основе ресурсов, которые могут использоваться при проведении атаки: (1) методов, которые могут получать и отправлять данные; (2) каналов, которые используются для передачи данных; (3) данных. Вклад каждого ресурса в значение показателя определяется при помощи показателя *отношение потенциал разрушений-усилия* (*Damage Potential-Effort Ratio*). Меньшие усилия (предусловия) и больший потенциал разрушений (постусловия) ведут к большему значению показателя. Показатель *поверхность атаки* напрямую связан с риском – чем больше поверхность атаки, тем больше риск компрометации системы.

1.3.6 Классификации показателей защищенности

Из предыдущих разделов видно, что существует огромное количество показателей защищенности, основанных на различных характеристиках объектов оценки защищенности. Спектр используемых показателей достаточно широк, методики их вычисления отличаются в различных работах, и нет единой системы их применения. Для упорядочивания различных показателей были созданы классификации показателей защищенности.

В ряде работ категории показателей выделяются согласно объектам оценки защищенности, например, техническая и организационная категории [139]. В [76] помимо этих двух категорий выделена категория управления. Классификация, предложенная институтом NIST [133] дополнительно включает 17 подкатегорий. В [128] в классификацию показателей добавлен дополнительный уровень, включающий три категории: безопасность, качество обслуживания и доступность. Для каждой из этих категорий определены технические показатели, организационные показатели и показатели управления. В [134], помимо разделения на категории показатели делятся по их функциям для бизнеса: управление инцидентами; управление уязвимостями; управление заплатками; управление конфигурациями; управление изменениями; безопасность приложений; финансовые показатели. По способу вычисления показатели делятся на первичные и вторичные [80]. Также выделяются показатели, вычисляемые на основе графов атак (вероятность атаки, уровень навыков атакующего и другие) и на основе графов зависимостей сервисов (ущерб от атаки/реагирования, эффективность реагирования и другие) [89]. В [49] выделяется 8 категорий показателей согласно типу значений показателей. На основе рассмотренных выше работ, можно классифицировать показатели по объекту оценки: показатели, относящиеся к конфигурационным характеристикам системы; показатели, характеризующие атаку; показатели, связанные с защитными мерами; показатели, характеризующие атакующего; показатели, характеризующие уровень защищенности системы в целом.

Современные КС содержат огромное количество информации, связанной с безопасностью. При оценке защищенности важно принимать во внимание различные характеристики объектов оценки. Обнаружить классификацию показателей, учитывающих различные объекты оценки защищенности, характеризующих текущее состояние ИС на различных уровнях детализации и с учетом разнообразных воздействующих факторов и возможных мер по реагированию на инциденты ИБ (контрмер), и применимую для оценки защищенности в SIEM-системах, в различных режимах работы системы (статическом и

динамическом), найти не удалось. В диссертационном исследовании предполагается решить эту проблему.

1.4 Требования к методикам оценки защищенности и выбора защитных мер

По результатам анализа текущей ситуации в области оценки защищенности и выбора защитных мер были сформулированы требования к методикам оценки защищенности и выбора защитных мер, в основу реализации которых должен быть положен модельно-методический аппарат, разрабатываемый в данной работе. Выделяются функциональные и нефункциональные требования. Функциональные требования определяют функции, которые должна выполнять система, реализующая разрабатываемые методики. Нефункциональные требования описывают требования и ограничения, налагаемые на ресурсы, потребляемые системой (например, временные ограничения, перечень используемых стандартов) [30, 37].

В соответствии с жизненным циклом ИС (КС) были выделены два режима работы разрабатываемой системы: статический (соответствует этапам проектирования и реализации ИС) и динамический (соответствует режиму эксплуатации ИС). Отличие статического режима: отсутствие жестких временных ограничений. В динамическом режиме время ограничено, и важно постоянно отслеживать и учитывать изменяющуюся ситуацию по безопасности на основе информации об инцидентах безопасности.

Функциональные требования к системе, реализующей разрабатываемые методики, были разделены на общие требования, требования, предъявляемые к системе в статическом режиме работы, и требования, предъявляемые к системе в динамическом режиме работы. Общие требования:

1) Система должна формировать адекватную ситуации и актуальную оценку защищенности КС и выбирать рациональные контрмеры на основе доступных входных данных в статическом и динамическом режимах.

2) Оценка защищенности должна быть представлена в виде комплекса показателей защищенности.

3) Комплекс показателей защищенности должен соответствовать последним наработкам в данной области.

4) Система должна учитывать характеристики атакующего, в том числе, его цели, положение в сети, первичные знания о сети, навыки и возможности по реализации атак.

5) Система должна учитывать связи между сервисами КС для учета распространения ущерба в случае успешной реализации атак, или побочного ущерба при реализации контрмер.

6) Система должна учитывать стоимостные характеристики атак и контрмер, чтобы определять выигрыш в случае реализации контрмер.

7) Процесс представления и обработки данных, применяемых для оценки защищенности и выбора защитных мер, должен быть автоматизирован.

8) Система должна выбирать рациональные технические контрмеры (снижающие уровень риска) с учетом стоимостных требований.

Требования, предъявляемые к системе в статическом режиме работы:

1) Система должна выявлять возможные атаки на КС, и характеризующий их набор показателей. Это позволит выявлять уязвимые места сети.

2) Система должна выявлять уязвимые места КС. Это необходимо для того, чтобы определять компоненты сети, для которых защитные меры необходимо внедрять в первую очередь.

3) Система должна формировать набор контрмер для повышения уровня защищенности системы.

Требования, предъявляемые к системе в динамическом режиме работы:

1) Система должна учитывать события безопасности, происходящие в КС, и производить переоценку ситуации по защищенности в соответствии с полученной информацией.

2) Система должна интегрироваться с SIEM-системами для учета предоставляемых ими данных. Это связано с широким применением SIEM-систем для обработки данных по безопасности в крупных ИС.

3) Система должна выбирать защитные меры, которые предотвратят развивающуюся атаку.

Задачей исследования является разработка системы оценки защищенности и выбора защитных мер, реализующей соответствующие методики. Предлагаемые методики будут основаны на комплексе показателей защищенности, которые позволят учитывать различные характеристики компьютерных атак, а также различные аспекты функционирования защищаемой системы. Подход подразумевает использование более сложных алгоритмов вычисления показателей при статическом режиме работы, учет информации о событиях безопасности в режиме, близком к реальному времени, а также быстрый перерасчет показателей на основе новой поступающей информации, что позволит отслеживать направление и уровень сложности атаки, цели и характеристики

атакующего. Кроме того, данная система позволит выбирать контрмеры для повышения общего уровня защищенности системы и для реагирования на отдельные атаки, выполняемые в реальном времени.

Нефункциональные требования можно разделить на группы: (1) своевременность; (2) обоснованность; (3) ресурсопотребление.

Своевременность — способность системы оценки защищенности и выбора защитных мер формировать результат в установленные сроки. Требование к своевременности задается в виде: $P_{СВ}(t \le T^{ДОП}) \ge P_{СВ}^{ДОП}$, где $P_{СВ}$ — вероятность своевременного получения результатов, t — время работы системы, $T^{ДОП}$ — допустимое время работы системы, $P_{СВ}^{ДОП}$ — допустимое значение вероятности.

Обоснованность – свойство соответствия результатов оценки защищенности и выбора защитных мер заданному состоянию КС. Требования к обоснованности являются основными в данной работе. Показатели обоснованности в статическом режиме: количество анализируемых сценариев атак; количество учитываемых параметров. Показатели обоснованности в динамическом режиме: точность выявления сценария атаки; финансовый выигрыш в случае реализации контрмер. Требования к количеству анализируемых сценариев атак и количеству учитываемых параметров задаются сравнением с существующими системами [37, 67, 74, 85, 91, 98, 125, 145]: $N_C \ge \max_{s \in S} N_C^s$, $N_П \ge \max_{s \in S} N_П^s$, где N_C — количество анализируемых сценариев атак, $N_П$ — количество учитываемых параметров предлагаемой системы оценки защищенности и выбора контрмер, S — множество существующих систем оценки защищенности и выбора контрмер, N_C^s — количество анализируемых сценариев атак, $N_П^s$ — количество учитываемых параметров существующей системы оценки защищенности и выбора контрмер s.

Точность выявления сценария атаки определяется на основе отклонения выявленного сценария от реального AC (чем меньше отклонение, тем выше точность): $AC \to \min$. Финансовый выигрыш в случае реализации контрмер определяется на основе индекса AL: $AL \to EL$.

Разрабатываемая система должна обнаруживать не меньше сценариев атак, чем существующие аналоги, и учитывать большее количество параметров. При этом точность выявления сценариев атаки и выигрыш в случае реализации контрмер должны

стремиться к максимуму. Это позволит говорить о том, что новая система превосходит существующие аналоги по функциональности и качеству анализа.

Ресурсопотребление отражает перечень и количество необходимых программных и аппаратных средств, объемы требуемых массивов информации, кадровые и другие ресурсы, необходимые для функционирования системы оценки защищенности и выбора контрмер. Требования к ресурсопотреблению задаются следующим образом: $P_{PEC}(r \le R^{ДОП}) \ge P_{PEC}^{ДОП}$, где P_{PEC} — вероятность того, что ресурсы, затрачиваемые на оценку защищенности и выбор контрмер r, не превышают допустимого значения $R_{ДОП}$, $P_{PEC}^{ДОП}$ — допустимое значение вероятности. Все заданные требования приведены в таблице 3.

Таблица 3 – Требования к системам оценки защищенности и выбора контрмер

Свойства	Показатели	Требования
Своевременность	Вероятность своевременного выполнения процесса оценки защищенности и выбора контрмер	$P_{CB}(t \le T^{ДОП}) \ge P_{CB}^{ДОП}$
Обоснованность	Количество анализируемых сценариев атак, количество учитываемых параметров, точность выявления сценария атаки, выигрыш в случае реализации контрмер	$N_C \ge \max_{s \in S} N_C^s$ $N_П \ge \max_{s \in S} N_П^s$ $AC \to \min$ $AL \to EL$
Ресурсопотребление	Вероятность того, что количество использованных ресурсов не превысит допустимое значение	$P_{PEC}(r \le R^{ДОП}) \ge P_{PEC}^{ДОП}$

1.5 Постановка задачи исследования

Система оценки защищенности и выбора контрмер должна выявлять «узкие» места КС, определять развитие атаки на основе имеющейся информации об инцидентах безопасности, и выбирать своевременные и адекватные контрмеры. Для удовлетворения заявленных требований предполагается использовать подход, основанный на комплексной системе показателей защищенности и соответствующих алгоритмах их вычисления, которые позволят учитывать различные характеристики компьютерных атак, а также различные аспекты функционирования защищаемой системы.

Во время работы система должна рассчитывать различные показатели защищенности, характеризующие текущую ситуацию наиболее полно на основе доступных данных. Для этого предполагается использовать различные модели, сформированные на основе входных данных: модель сети, модель атак, модель

зависимостей сервисов, модель атакующего, модель события безопасности, модель контрмеры.

Результаты работы системы в статическом режиме: набор показателей, характеризующих уровень защищенности КС, и набор контрмер, которые позволят повысить уровень защищенности сети. В динамическом режиме это: набор показателей, характеризующих обнаруженную атаку, и адекватные контрмеры.

Задачу разработки системы оценки защищенности и выбора контрмер можно разделить на подзадачи:

- Разработка комплекса показателей защищенности с учетом различных входных данных, таких как модели КС, атакующих действий, атакующих, инцидентов безопасности и контрмер, и на различных уровнях функционирования защищаемой системы.

- Разработка методики оценки защищенности КС с учетом доступных данных.

- Разработка и модификация моделей, необходимых для вычисления показателей защищенности.

- Разработка алгоритмов вычисления показателей защищенности на различных уровнях системы показателей.

- Разработка методики выбора контрмер для повышения уровня защищенности КС и реагирования на компьютерные атаки с учетом доступных данных и требований бизнеса.

- Построение архитектуры системы и реализация программного средства для оценки защищенности КС и выбора защитных мер для повышения уровня защищенности КС и реагирования на компьютерные атаки на основе предложенных методик.

Взаимосвязь подзадач показана на рисунке 6.

На содержательном уровне научную задачу диссертационного исследования можно сформулировать следующим образом: разработать модельно-методический аппарат (комплекс моделей, методик и алгоритмов), реализующий оценку защищенности КС и выбор контрмер для SIEM-систем. Реализация таких моделей, методик и алгоритмов в системах оценки защищенности и выбора контрмер должна позволить снизить негативный эффект атак на КС.

Для оценки защищенности и выбора контрмер необходимо разработать модели: модель сети, модель зависимостей сервисов, модель атакующего, модель атак, модель события безопасности, модель защитной меры, и методики: методику оценки защищенности и методику выбора защитных мер, и входящие в их состав алгоритмы.

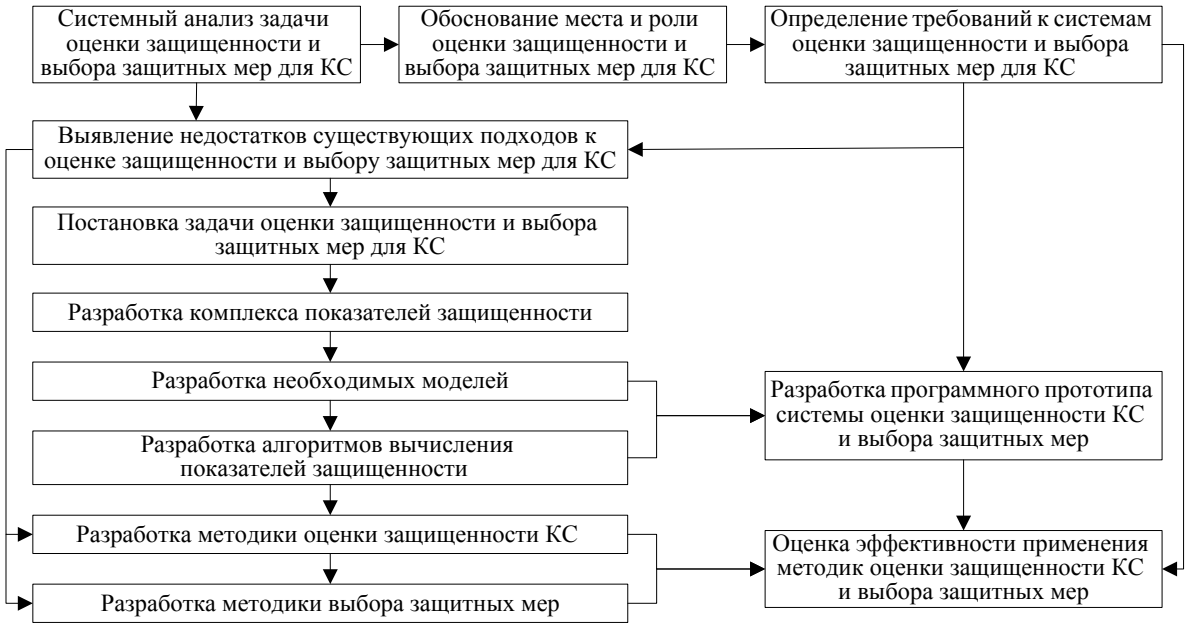


Рисунок 6 – Схема проведения диссертационного исследования

Входными данными методики оценки защищенности КС являются: N – модель КС; S – модель зависимостей сервисов; A – модель атак; M – модель атакующего; E – модель событий. Необходимыми для оценки являются как минимум модели N и S. Остальные модели являются дополнительными и позволяют получить более точную оценку. Модель E отвечает за оценку защищенности в динамическом режиме.

Входными данными методики выбора защитных мер являются: комплекс показателей защищенности (результат работы методики оценки защищенности) и модели R защитных мер.

Пусть R_a – риск успешной реализации атаки $a \in A$ на КС, где A – множество всех атак на КС. R_a является результатом работы функции определения риска атаки $RiskCalc(a)$. Обозначим как C – множество средств защиты и защитных мер, рекомендуемых системой оценки защищенности и выбора защитных мер для атаки a. Тогда результат работы функции $RiskCalc(a, C)$ – риск успешной реализации атаки a в случае реализации защитных мер C. Целевой функцией методики оценки защищенности и выбора контрмер является снижение риска в случае атак на КС (поскольку целью исследования является повышение уровня защищенности КС, а уровень защищенности КС тем выше, чем ниже риск компрометации КС) $RiskCalc(a, C) \rightarrow \min$ для всех a, при соблюдении требований к остальным свойствам системы оценки защищенности и выбора защитных мер:

к своевременности: $P_{СВ}(t \le T^{ДОП}) \ge P_{СВ}^{ДОП}$, где $P_{СВ}^{ДОП} = 0{,}99$ (значение определено в соответствии со стандартными требованиями к подобным системам) [37, 43]; $T^{ДОП}$ –

допустимое время проведения оценки и выбора защитных мер, $T^{ДОП} = T_{пр}^{TP}$, где при оценке защищенности и выборе защитных мер в статическом режиме $T_{пр}^{TP}$=1 мин. (время было выбрано на основе результатов опроса экспертов и анализа существующих работ [37, 43, 91, 98, 105], таблица 4, а в динамическом режиме – $T_{пр}^{TP}$=10 секунд (время было выбрано на основе анализа существующих работ [37, 43, 91, 98, 105]);

Таблица 4 – Время оценки защищенности и выбора контрмер в существующих исследованиях

Операции\ исполнитель	Система оценки защищенности [37]	Система оценки защищенности [43]	Система оценки защищенности NetSPA [98]	Система оценки защищенности и выбора контрмер [91]
Оценка защищенности сети и генерация рекомендаций	45 мин. в режиме проектирования и 25 мин. в режиме эксплуатации для сети из 40 хостов (генерация общих рекомендаций).	2 мин. в режиме проектирования и 10 сек. в режиме эксплуатации (изменено более 10% хостов) для сети из 1000 хостов.	1 мин. в режиме проектирования для анализа графа и выбора контрмер для сети до 2000 хостов. (генерация общих рекомендаций).	7 сек. в режиме эксплуатации для анализа графа и выбора контрмер для сети из 800 хостов.

к обоснованности: $N_C \geq \max_{s \in S} N_C^s$, $N_П \geq \max_{s \in S} N_П^s$, $AC \to \min$, $AL \to EL$, где N_C — количество анализируемых сценариев атак, $N_П$ — количество учитываемых параметров предлагаемой системы оценки защищенности и выбора защитных мер, S — множество существующих систем оценки защищенности и выбора защитных мер, N_C^s — количество анализируемых сценариев атак, а $N_П^s$ — количество учитываемых параметров существующей системы оценки защищенности и выбора защитных мер s, AC – индекс отклонения сценариев атаки, AL — индекс выигрыша в случае реализации контрмер. В таблице 5 приведены параметры, применяемые для оценки защищенности и выбора контрмер в существующих системах оценки защищенности. На основе таблицы 5 и анализа релевантных работ было сформировано множество учитываемых параметров: конфигурация оцениваемой сети (программно-аппаратное обеспечение, включая зависимости между сервисами), параметры атакующего (навыки, положение в сети, цели, первоначальные знания о сети), модель атак, параметры событий безопасности, модель защитной меры. Все модели используются для получения показателей защищенности, максимально точно отражающих текущую ситуацию;

к ресурсопотреблению: $P_{РЕС}(r \leq R^{ДОП}) \geq P_{РЕС}^{ДОП}$, где $P_{РЕС} = 0,99$, $R^{ДОП} = 0,75$ (75% от общего ресурса, доступного приложениям, так как часть ресурсов отводится ОС) для ресурсов: оперативная память, жесткий диск, процессорное время.

Таблица 5 – Параметры, применяемые для оценки защищенности и выбора контрмер в существующих системах

Система	Параметр								
	Базовые показатели (множество дополнительных показателей, рассчитываемых на основе параметров КС)	Характеристики атаки (вероятность)	Характеристики атакующего (уровень навыков, положение, мотивация)	Ущерб от атаки (потери)	Общая оценка защищенности (уровень риска или уровень защищенности, поверхность атаки)	Затраты на контрмеры	Побочный ущерб от реализации контрмер	Эффективность контрмер	События безопасности
Poolsappasit et al. [125]	-	+ Байесовский граф атак	-	+- Устанавливается вручную	+ Ожидаемые потери	+- Устанавливается вручную	+- Устанавливается вручную	+	+
Dantu et al. [67]	-	+- Байесовская сеть доверия на основе профиля атакующего	+ Реалистичный профиль атакующего на основе набора характеристик	-	+ Вероятность атаки	-	-	+ Изменение характеристик атакующего, необходимых для компрометации узла	-
Kheir et al. [89]	-	-	-	+ На основе зависимостей сервисов	-	+- Устанавливается вручную	+ На основе зависимостей сервисов	+ На основе зависимостей сервисов	-
Wu et al. [145]	-	+- Граф атак на основе достижимости сервиса	-	+ На основе зависимостей сервисов и исторических данных о событиях безопасности	-	-	+ Результаты атаки после реализации контрмер	+ Результаты атаки после реализации контрмер	+
Степашкин [37]	-	+ Граф атак	+- Позиция атакующего и привилегии в сети	+- Качественные оценки критичности ресурсов	+ Ущерб от атаки и вероятность атаки	-	-	-	-
Jahnke et al. [85]	-	+- Граф доступности	-	+- Изменение доступности ресурса	-	-	+- Изменение доступности ресурса	+- Изменение доступности ресурса	+- Устанавливается вручную
Lippmann et al. [98]	-	+ Граф атак	-	+- Количество хостов для которых получен доступ администратора	-	-	-	+- Снижение количества скомпрометированных хостов	-
Granadillo et al. [74]	-	+ Поверхность атаки	-	+ Затронутый ресурс системы	+ Поверхность атаки	-	+ Объем ресурсов, не затронутых атакой	+ Снижение объема атаки	-

Результат работы методики оценки защищенности КС: комплекс показателей, характеризующих состояние защищенности КС. Результат работы методики выбора защитных мер в статическом режиме: набор средств защиты, позволяющих повысить уровень защищенности КС. Результат работы методики выбора защитных мер в динамическом режиме: набор защитных мер для предотвращения развития атаки.

Выводы по главе 1

В данном исследовании предполагается сосредоточиться на процессах оценки и обработки риска. На основе анализа существующих стандартов в области менеджмента ИБ можно сделать вывод, что для эффективной обработки риска предпочтительной является детальная количественная оценка риска, которая включает тщательное определение и установление ценности активов, оценку угроз этим активам и оценку уязвимостей. Это достаточно сложные и трудоемкие процессы, требующие серьезных вложений. В современных организациях даже небольшое время нарушения корректного функционирования ИТ систем может привести к серьезным финансовым потерям, а своевременное обнаружение атакующего и принятие защитных мер может их предотвратить. Однако, из-за сложности и размеров современных ИТ инфраструктур, процесс оценки рисков может занять значительное время. Поэтому важной задачей является определение методик количественной оценки и обработки риска, и автоматизация данных процессов, что в данном исследовании предлагается решить на основе автоматизированного анализа моделей предметной области, полученных путем аналитического моделирования, с использованием существующих стандартов унифицированного представления элементов оценки и обработки риска.

Современные SIEM-системы предоставляют возможности по эффективному управлению ИБ организаций, однако реализованные в них методики оценки рисков не дают полной картины информационных и бизнес-рисков, порождаемых потенциальными угрозами, и не позволяют принять всесторонне обоснованное решение по выбору и внедрению защитных мер. Поэтому существует необходимость в создании автоматизированного подхода, основанного на объективных количественных показателях и позволяющего оценить уровень защищенности системы, выявить ее слабые места и выбрать эффективные защитные меры.

Формальная постановка задачи определяет построение методики оценки защищенности и выбора защитных мер как формирование набора показателей, адекватно отражающих ситуацию по защищенности, и защитных мер, позволяющих минимизировать финансовые потери при проведении атак на КС, при соблюдении требований к свойствам своевременности, обоснованности и ресурсопотребления.

Глава 2 Комплекс показателей защищенности компьютерных сетей. Алгоритмы расчета показателей защищенности. Методики оценки защищенности и выбора защитных мер

2.1 Комплекс показателей защищенности компьютерных сетей

Выделение и обоснование учитываемых характеристик. Чтобы оценить риски, вначале необходимо их идентифицировать (рисунок 3). Перспективным подходом в этой области является аналитическое моделирование атак в виде графов. Данный подход был подробно рассмотрен в диссертациях [37, 43]. На рисунке 7 представлена связь между подходами, предложенными в [37, 43] и данным исследованием. Уровень идентификации риска выходит за пределы исследования. Остальные уровни соответствуют области исследования. Из рисунка видно, что результатом оценки защищенности является набор показателей защищенности.

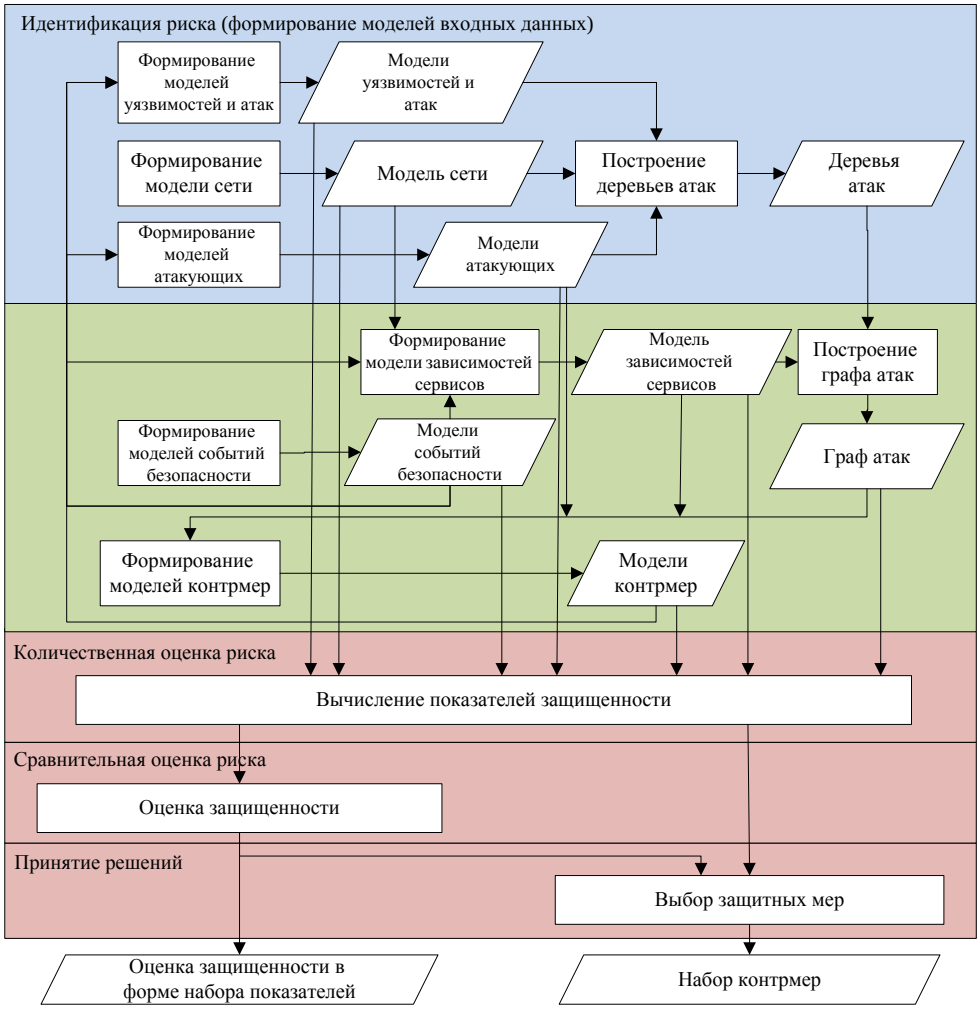


Рисунок 7 – Связь между методами аналитического моделирования и оценкой защищенности и выбором защитных мер

Основными компонентами оценки защищенности КС являются: активы, источники угроз, угрозы, уязвимости, воздействия, защитные меры (контрмеры) и риск. В исследовании модели этих компонентов используются в качестве входных данных. Применяются как модели, предложенные в [43] (скорректированные в соответствии с требованиями исследования), так и ряд новых моделей. Используемые в качестве входных данных компоненты анализа защищенности стали основой для формирования предлагаемого в данном исследовании комплекса показателей защищенности.

В главе 1 были рассмотрены существующие в настоящий момент исследования в области показателей защищенности, а также попытки в области их классификации. В данной работе показатели защищенности являются основой методик оценки защищенности и выбора контрмер.

Для поставленных в диссертационном исследовании задач показатели защищенности и их классификация должны удовлетворять требованиям: классификация должна соответствовать последним исследованиям в области показателей защищенности; показатели должны рассчитываться на основе подхода к анализу защищенности, применяющего в качестве входных данных графы атак и графы зависимостей сервисов; показатели должны позволять получить оценку текущей ситуации по защищенности в любой момент времени на основе доступных входных данных; показатели должны учитывать информацию, поступающую от SIEM-систем; показатели должны позволять выбрать контрмеры в любой момент времени на основе доступных входных данных; классификация должна позволять выделить статический и динамический режимы оценки защищенности и выбора контрмер; классификация должна позволять оценить стоимостные характеристики атак и контрмер.

Найти классификацию, удовлетворяющую данным требованиям, не удалось. Поэтому была разработана своя классификация показателей защищенности. Были выделены следующие уровни классификации: топологический уровень; уровень графа атак; уровень атакующего; уровень событий; уровень выбора контрмер и уровень системы (интегральный уровень). Каждая категория включает следующие подкатегории: базовые характеристики, стоимостные характеристики, характеристики нулевого дня. Хотя характеристики нулевого дня выделены в отдельную подкатегорию, алгоритмы их вычисления выходят за рамки данного исследования. Внутри каждой подкатегории

выделяются основные показатели, используемые для вычисления значения уровня риска, и вспомогательные, не используемые для вычисления значения уровня риска.

Для удовлетворения требований адекватного отражения текущей ситуации на основе доступных входных данных и применения в качестве входных данных графов атак и графов зависимостей сервисов, уровни классификации выделены в зависимости от обрабатываемых входных данных: знаний о сети, ее уязвимостях, атакующем, или происходящих в системе событиях безопасности. Для удовлетворения требования интеграции с SIEM-системами и выделения статического и динамического режимов оценки защищенности и выбора защитных мер введен уровень событий, принимающий в качестве входных данных информацию о событиях безопасности. Для учета контрмер при оценке защищенности, а также выбора контрмер, добавлен уровень выбора контрмер. Для учета стоимостных характеристик атак и контрмер введена категория стоимостных характеристик. Каждый уровень содержит набор показателей, отражающий последние исследования в данной области в соответствии с обзором, приведенным в главе 1.

На рисунке 8 представлена схема разработанного комплекса показателей. Для вычисления интегральных показателей достаточно показателей топологического уровня. В дальнейшем значения интегральных показателей можно уточнять показателями остальных уровней, что показано пунктирными стрелками.

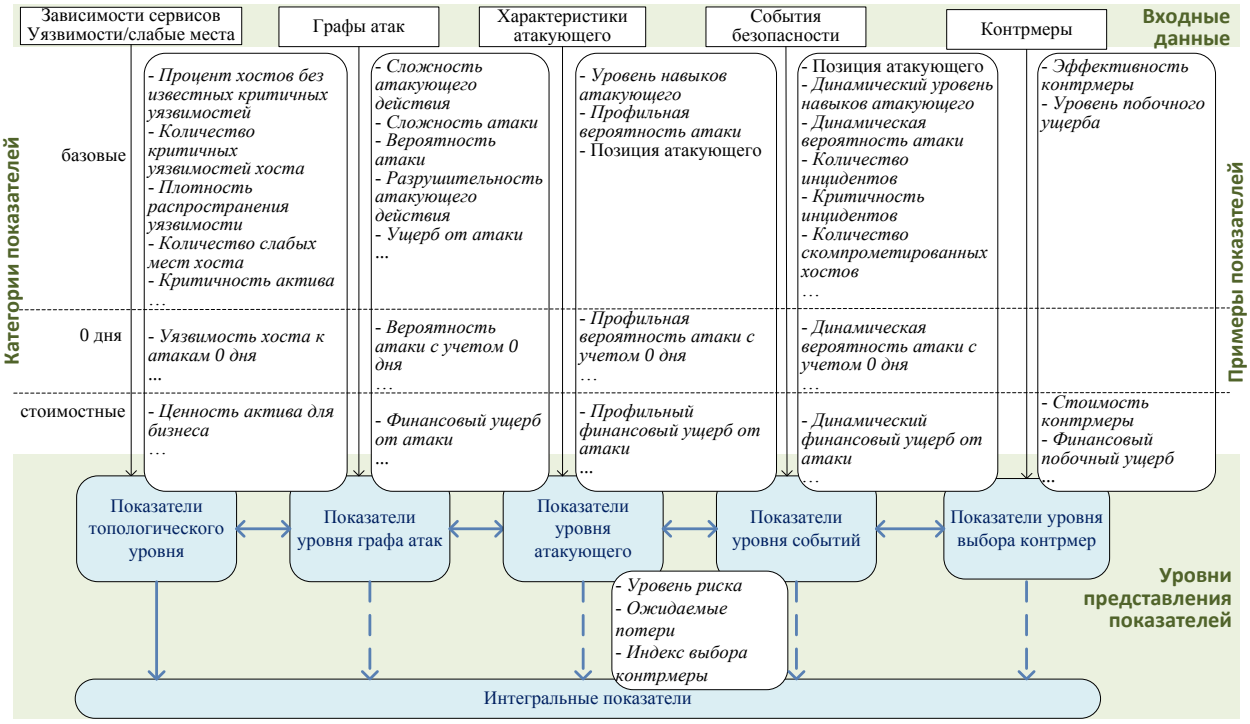


Рисунок 8 – Комплекс показателей защищенности

Описание показателей каждой группы. В литературе даются различные определения показателей защищенности. В данном исследовании под показателем защищенности будем понимать систему взаимосвязанных измерений, позволяющих квантифицировать отдельную характеристику, то есть, это измерение, которое сравнивается со шкалой или критерием для получения значимого результата [55, 108]. Показатели защищенности необходимы для отслеживания ситуации по защищенности и облегчения выполнения задач по улучшению ситуации путем применения соответствующих действий [133]. Показатели являются результатом анализа (в отличие от измерений) и для выполнения поставленных целей они должны удовлетворять следующим требованиям (помимо функциональных требований, поставленных в главе 1) [55]: показатели должны быть ценными для целевой аудитории; стоимость измерений не должна превышать ценность показателей; своевременность и частота измерений должна соответствовать частоте изменений объекта измерений; показатели должны быть объективными и количественно выраженными; показатели должны быть воспроизводимыми разными экспертами в одинаковых условиях.

На первом уровне, *топологическом*, для вычисления показателей доступны входные данные: модель КС [43]; информация о хостах; характеристики программно-аппаратного обеспечения, в том числе уязвимости; характеристики уязвимостей на основе открытой базы уязвимостей NVD [119]; характеристики слабых мест хоста на основе базы слабых мест CWE [62]; сервисы КС; модель зависимостей сервисов.

Исходя из особенностей входных данных, используемых в данном исследовании, и на основе показателей из [134], были выбраны следующие вспомогательные базовые показатели: *процент хостов без известных критичных уязвимостей* (*Percent of Hosts with No Known Severe Vulnerabilities, PHNKSV*); *количество критичных уязвимостей хоста* (*Number of Known Severe Host Vulnerability Instances, NKSHVI*); *нормированное количество критичных уязвимостей хоста* (*Normalized Number of Known Severe Host Vulnerability Instances, NNKSHVI*), *плотность распространения уязвимости* (*Target Vulnerability Distribution, TVD*). И добавлены показатели: *количество слабых мест хоста* (*Number of Known Host Weakness Instances, NKHWI*); *нормированное количество слабых мест хоста* (*Normalized Number of Known Host Weakness Instances, NNKHWI*). Вспомогательные показатели 0 дня: *уязвимость хоста к атакам 0 дня* (*Host Vulnerability to Zero-Day Attacks*).

Были выбраны следующие основные базовые показатели: *критичность актива* (*Asset Criticality*); *разрушительность атакующего действия для свойств конфиденциальности/целостности и доступности* (*ConfImpact, IntegImpact, AvailImpact*); *ущерб, наносимый атакующим действием* (*Attack Impact*); *вероятность атакующего действия*. Основные стоимостные показатели: *ценность актива для бизнеса* (*Business Value of the Asset*); *финансовый ущерб* (*Monetary Impact*).

На уровне *графа атак* для вычисления показателей защищенности доступны входные данные: показатели защищенности и входные данные предыдущего *топологического* уровня; граф атак.

Исходя из особенностей входных данных, были выбраны следующие вспомогательные базовые показатели: *нормализованное количество атак, проходящих через хост* (*Quantified Number of Attacks that Go through the Host*); *нормализованное количество атак с высоким уровнем риска, проходящих через хост* (*Quantified Number of Attacks through the Host with "High" RiskLevel*). Вспомогательные показатели 0 дня: *устойчивость сети к уязвимостям 0 дня*.

Были выбраны следующие основные базовые показатели: *сложность атакующего действия* (*Complexity of Attack Action*); *сложность атаки* (*Attack Complexity*); *вероятность атаки* (*Attack Potentiality*); *разрушительность атакующего действия*; *ущерб от атаки* (*Attack Impact*). Основные стоимостные показатели: *финансовый ущерб от атаки* (*Monetary Attack Impact*). Основные показатели 0 дня: *вероятность атаки с учетом уязвимостей 0 дня* (*Attack Potentiality Considering Zero-Days*).

На уровне *атакующего* для вычисления показателей защищенности доступны входные данные: показатели защищенности и входные данные предыдущих уровней; модель атакующего.

Исходя из особенностей входных данных, были выбраны следующие основные базовые показатели: *уровень навыков атакующего* (*Attacker Skill Level*); *позиция атакующего на графе атак* (*Attacker Position*); *профильная вероятность атаки* (*Profiled Attack Potentiality*). Основные стоимостные показатели: *профильный финансовый ущерб от атаки* (*Profiled Monetary Attack Impact*). Основные показатели 0 дня: *профильная вероятность атаки с учетом уязвимостей 0 дня* (*Profiled Attack Potentiality Considering Zero-Days*).

На уровне *событий* для вычисления показателей защищенности доступны входные данные: показатели защищенности и входные данные предыдущих уровней; модель события.

Были выбраны следующие вспомогательные базовые показатели: *количество инцидентов* (*Number of Incidents*). Данный показатель основан на показателе CIS *количество инцидентов* [134] и модифицирован для реализуемой системы следующим образом: *количество инцидентов на хосте* (*Number of Host Incidents*); *нормированное количество инцидентов на хосте* (*Normalized Number of Host Incidents*); *количество инцидентов в системе* (*Number of System Incidents*). И добавлены показатели: *средняя критичность инцидентов на хосте* (*Mean Criticality of Host Incidents*) и *средняя критичность инцидентов в системе* (*Mean Criticality of System Incidents*); *количество скомпрометированных хостов* (*Compromised Hosts*).

Были выбраны следующие основные базовые показатели: *динамический уровень навыков атакующего* (*Dynamic Attacker Skill Level*); *позиция атакующего на графе атак* (*Attacker Position*); *динамическая вероятность атаки* (*Dynamic Attack Potentiality*). Основные стоимостные показатели: *динамический финансовый ущерб от атаки* (*Dynamic Monetary Attack Impact*). Основные показатели 0 дня: *динамическая вероятность атаки с учетом уязвимостей 0 дня* (*Dynamic Attack Potentiality Considering Zero-Days*).

На уровне *выбора контрмер* для вычисления показателей защищенности доступны входные данные: показатели защищенности и входные данные предыдущих уровней; контрмеры.

Были выбраны следующие основные базовые показатели: *эффективность контрмеры* (*Countermeasure Effectiveness, CE*); *уровень побочного ущерба* (*Collateral Damage, CD*). Основные стоимостные показатели: *стоимость контрмеры* (*Countermeasure Cost, CC*); *финансовый побочный ущерб* (*Monetary Collateral Damage*).

На *интегральном* уровне для вычисления показателей защищенности доступны входные данные: показатели защищенности и входные данные предыдущих уровней.

Были выбраны следующие показатели: *уровень риска атаки* (*Attack Risk Level*); *уровень риска сервиса* (*Service Risk Level*); *уровень риска хоста* (*Host Risk Level*); *уровень риска КС* (*System Risk Level*); *индекс выбора контрмеры* (*Countermeasure Selection Index*). Стоимостные показатели: *ожидаемые потери* (*Loss Expectancy*).

2.2 Методика оценки защищенности компьютерных сетей

В разделе 1.4 были сформулированы основные требования к разрабатываемым методикам оценки защищенности КС и выбора контрмер. Для удовлетворения этих требований была разработана методика оценки защищенности КС на основе комплекса показателей защищенности. Обобщенная схема методики представлена на рисунке 9. Методика включает три этапа: (1) сбор входных данных; (2) вычисление показателей защищенности; (3) определение уровня защищенности.

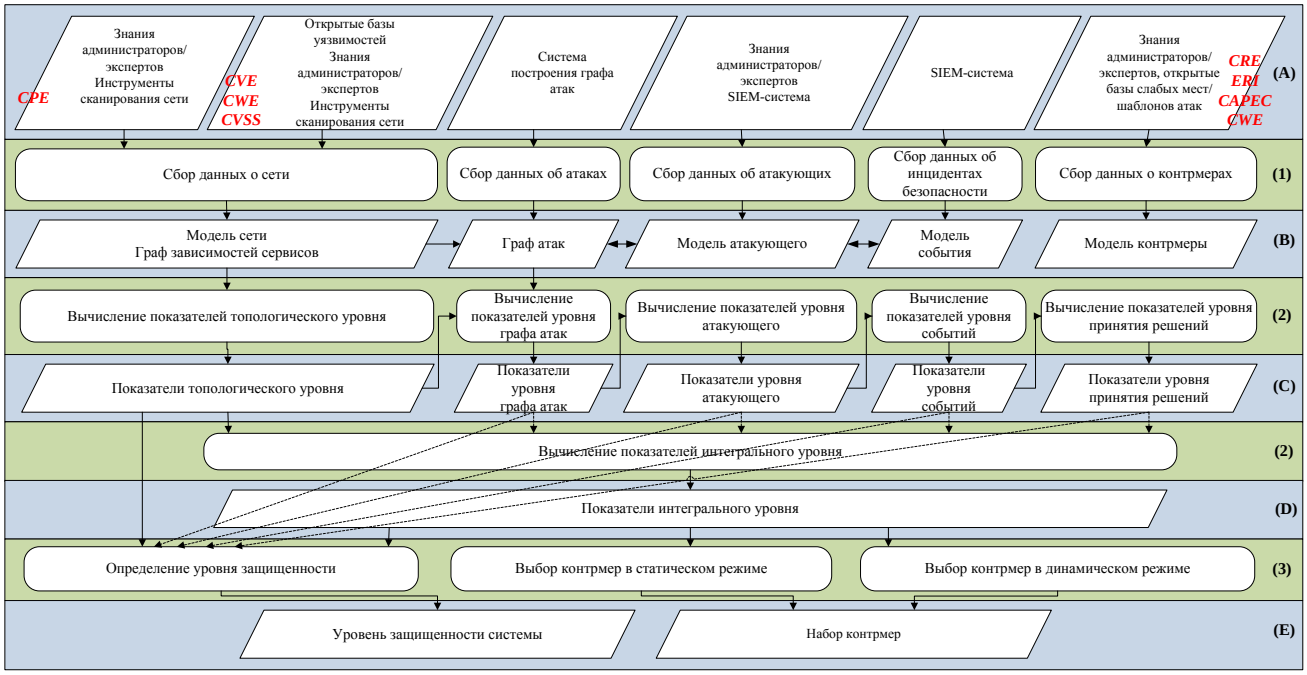


Рисунок 9 – Обобщенная схема подхода к оценке защищенности и выбору защитных мер

Полученные на каждом этапе выходные данные используются как входные данные следующего этапа. Перемещение данных между этапами показано стрелками. Верхний уровень A рисунка 9 соответствует входным данным, применяемым для вычисления показателей. Жирным курсивом выделены названия стандартов, применяемых для представления входных данных. Стандарты представления входных данных, входящие в протокол SCAP (подробно описанный в главе 1), применяются для удовлетворения требования автоматизации оценки защищенности и выбора защитных мер. Стандарт CVE используется для представления уязвимостей, CPE — для представления ПО, CCE — для представления конфигураций, CRE и ERI — для представления контрмер. Стандарт CVSS используется для оценивания уязвимостей. На основе собранных данных строятся модели, которые потом применяются для

вычисления показателей (уровень B). Алгоритмы вычисления показателей подробно рассмотрены в следующем разделе. Выходные данные отображены на уровне E. На уровнях C и D представлены данные, применяемые на соответствующих этапах подхода. Данные разделены на группы в зависимости от источников, в соответствии с уровнями системы показателей. Это позволяет выделить статический и динамический режимы работы методики. Статическому режиму работы соответствуют топологический уровень, уровень графа атак и атакующего. Более детально этапы работы методики в статическом режиме и последовательность их выполнения представлены на рисунке 10.

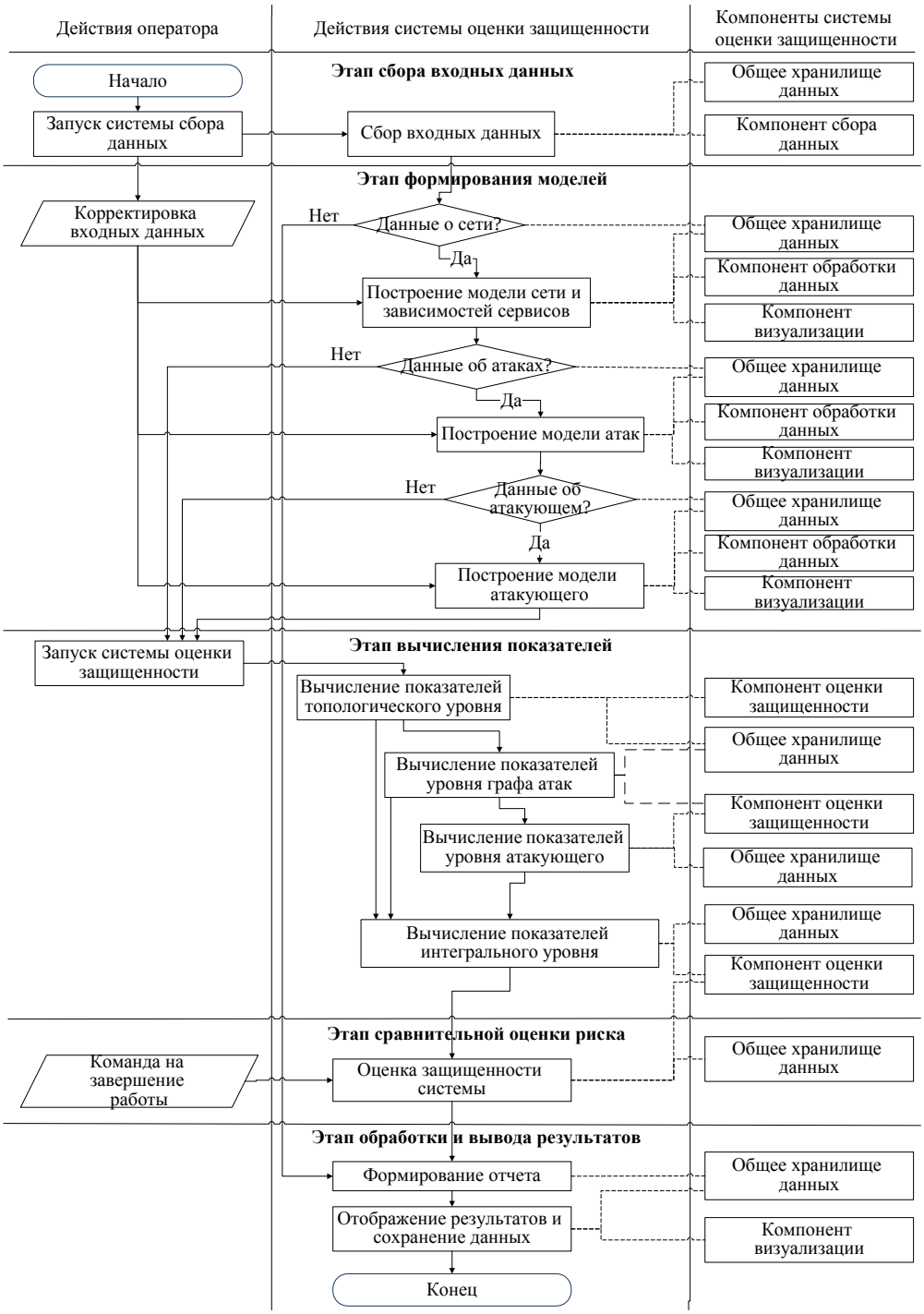


Рисунок 10 – Схема методики оценки защищенности в статическом режиме

Результат работы методики в статическом режиме: уровень риска КС и комплекс показателей защищенности. Для динамического режима работы дополнительно введен уровень событий. Более детально этапы работы методики в динамическом режиме и последовательность их выполнения представлены на рисунке 11. Результат работы в динамическом режиме: путь атаки, цель атаки, характеристики атакующего, ожидаемые потери в случае успешной реализации атаки (уровень риска).

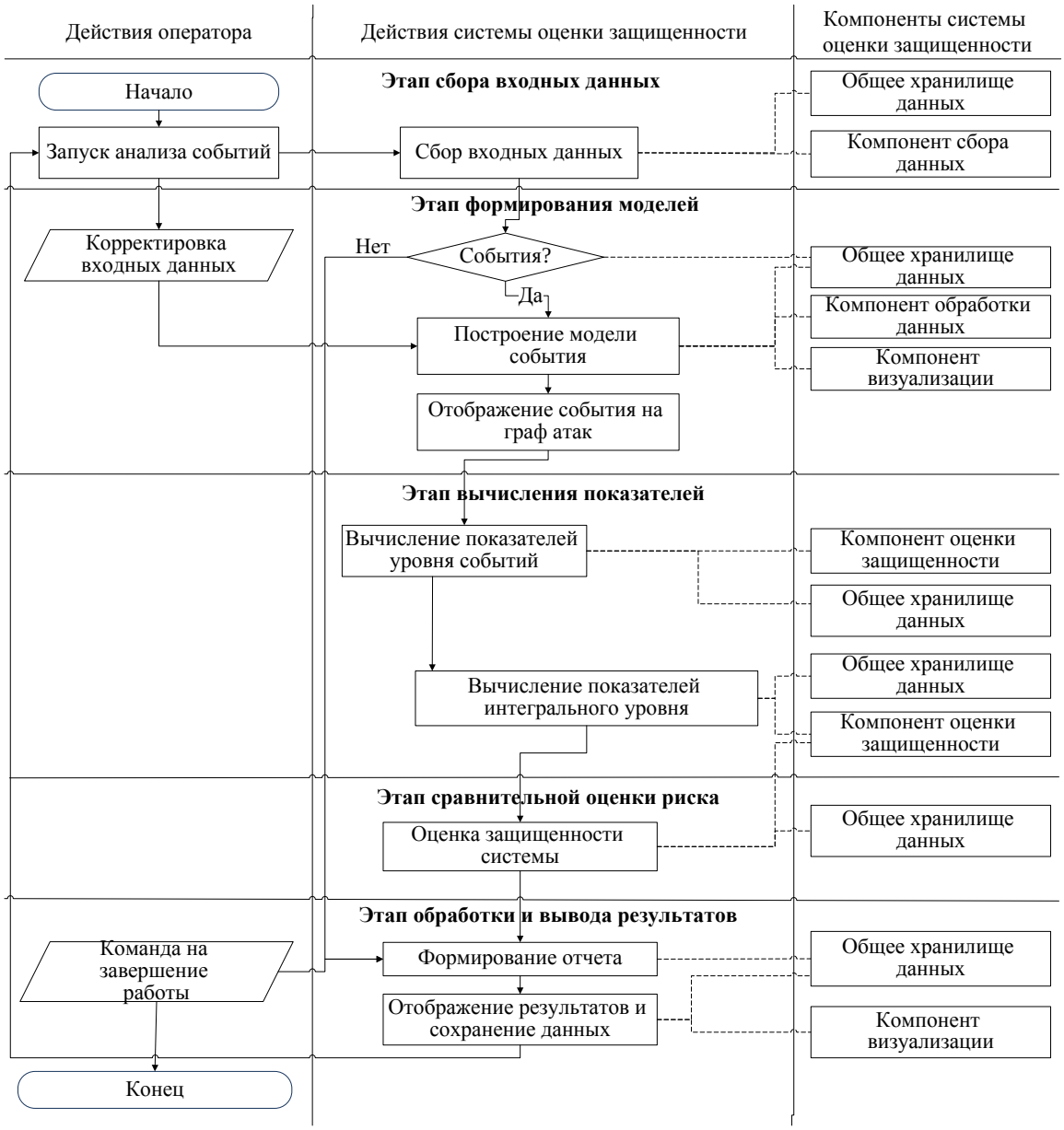


Рисунок 11 – Схема методики оценки защищенности в динамическом режиме

Подход относится к так называемым «any-time» подходам. Основной особенностью подхода является возможность получения оценок и выбора защитных мер на каждом уровне системы показателей. При получении новых данных (других уровней) оценки могут быть улучшены применением алгоритмов соответствующего уровня.

Общие показатели оценки защищенности и выбора защитных мер являются показателями интегрального уровня. Для их вычисления необходимы показатели хотя бы топологического уровня (все остальные уровни являются дополнительными, что показано на рисунке 9 пунктирными стрелками).

2.3 Алгоритмы вычисления показателей защищенности

В данном разделе рассматриваются алгоритмы вычисления основных и вспомогательных показателей, реализуемые на соответствующих этапах методики оценки защищенности КС.

Вычисление вспомогательных топологических показателей. Для вычисления вспомогательных топологических показателей применяются данные о хостах КС, уязвимостях их программно-аппаратного обеспечения, и характеристики уязвимостей, полученные из базы NVD [119].

Показатель *процент хостов без известных критичных уязвимостей PHNKSV* основан на показателе CIS *процент систем без известных критичных уязвимостей* (*Percent of Systems with No Known Severe Vulnerabilities*) [119], который был переопределен для хостов. Данный показатель отражает относительную уязвимость системы. Критичными в данном случае называются уязвимости со значением базовой оценки CVSS «Высокая». Показатель измеряется в процентах на шкале от 0 до 100 и может изменяться во времени. Формула для вычисления показателя:

$$PHWKSV = \frac{HWKSV}{n} \cdot 100,$$

где $HWKSV$ — количество хостов без известных критичных уязвимостей,

n — общее количество хостов в сети.

Показатели *количество критичных уязвимостей хоста NKSHVI* и *нормированное количество критичных уязвимостей хоста NNKSHVI* основаны на показателе CIS *количество известных уязвимостей* (*Number of Known Vulnerability Instances*) [119]. Критичными в данном случае называются уязвимости со значением базовой оценки CVSS «Высокая». Показатель *количество критичных уязвимостей хоста* может изменяться во времени.

Нормированное количество критичных уязвимостей хоста отображает количество критичных уязвимостей хоста по отношению к другим хостам и может изменяться во времени:

$$NNKSHVI = \frac{NKSHVI_i}{\max_i NKSHVI_i},$$

где $NKSHVI_i$ — количество критичных уязвимостей *i*-го хоста;

$max_i NKSHVI_i$ — максимальное количество критичных уязвимостей по всем хостам сети.

Показатель *количество слабых мест хоста NKHWI* может изменяться во времени. Данный показатель рассчитывается на основе слабых мест программно-аппаратного обеспечения хоста в соответствии со словарем CWE [62]. Показатель *нормированное количество слабых мест хоста NNKHWI* отображает количество слабых мест хоста по отношению к другим хостам и может изменяться во времени:

$$NNKHWI = \frac{NKHWI_i}{\max_i NKHWI_i},$$

где $NKHWI_i$ — количество слабых мест *i*-го хоста;

$max_i NKHWI_i$ — максимальное количество слабых мест по всем хостам сети.

Показатель *плотность распространения уязвимости TVD* основан на контекстном показателе CVSS *плотность целей* (*Target Distribution*). Для расчета данного показателя для уязвимости *v* предлагается использовать формулу:

$$TVD_v = \frac{n\_vuln}{n} \times 100,$$

где n_vuln — количество хостов сети, на которых обнаружена уязвимость v;

n — общее количество хостов в сети.

Показатель измеряется в процентах и может использоваться для ранжирования уязвимостей системы по уровню их распространения.

Алгоритмы вычисления основных топологических показателей. Для вычисления уровня риска используется определение, данное в стандарте [6], согласно которому риск характеризуется комбинацией вероятности возникновения инцидента (или проведения атаки) и его разрушительного воздействия. На основе анализа, приведенного в главе 1, можно сделать вывод, что последствия в случае успешной реализации атаки (разрушительное воздействие инцидента) зависят от ценности

(критичности) актива для его владельцев и разрушительности атакующего действия. Успешность атаки (вероятность возникновения инцидента) зависит от наличия уязвимости, которая позволит осуществить атаку, наличия доступа к этой уязвимости, сложности ее эксплуатации, возможностей атакующего, и привлекательности уязвимости (то есть ее важности для достижения цели атакующего). Поэтому в качестве основных были выбраны следующие показатели: *критичность актива*; *разрушительность атакующего действия*; *ущерб от атакующего действия*; *вероятность атакующего действия*. Стоимостные показатели, используемые для вычисления значения уровня риска: *ценность актива для бизнеса* (*Business Value of the Asset*); *финансовый ущерб* (*Monetary Impact*).

Алгоритм оценки критичности активов ИС. Критичность актива определяет важность конкретного актива для целей и миссии организации. Критичность актива измеряется по шкале от 0 до 100. Для оценки критичности активов предлагается объединить операции организационного уровня, описанные в стандарте [10], и технического уровня, предлагаемые в ряде исследовательских работ [89, 131, 136]. То есть перейти от целей и миссии организации к информационным активам, которые их поддерживают, и связанным с ними угрозам и уязвимостям. Для этого выделим два этапа алгоритма: этап организационного уровня и этап технического уровня (рисунок 12).

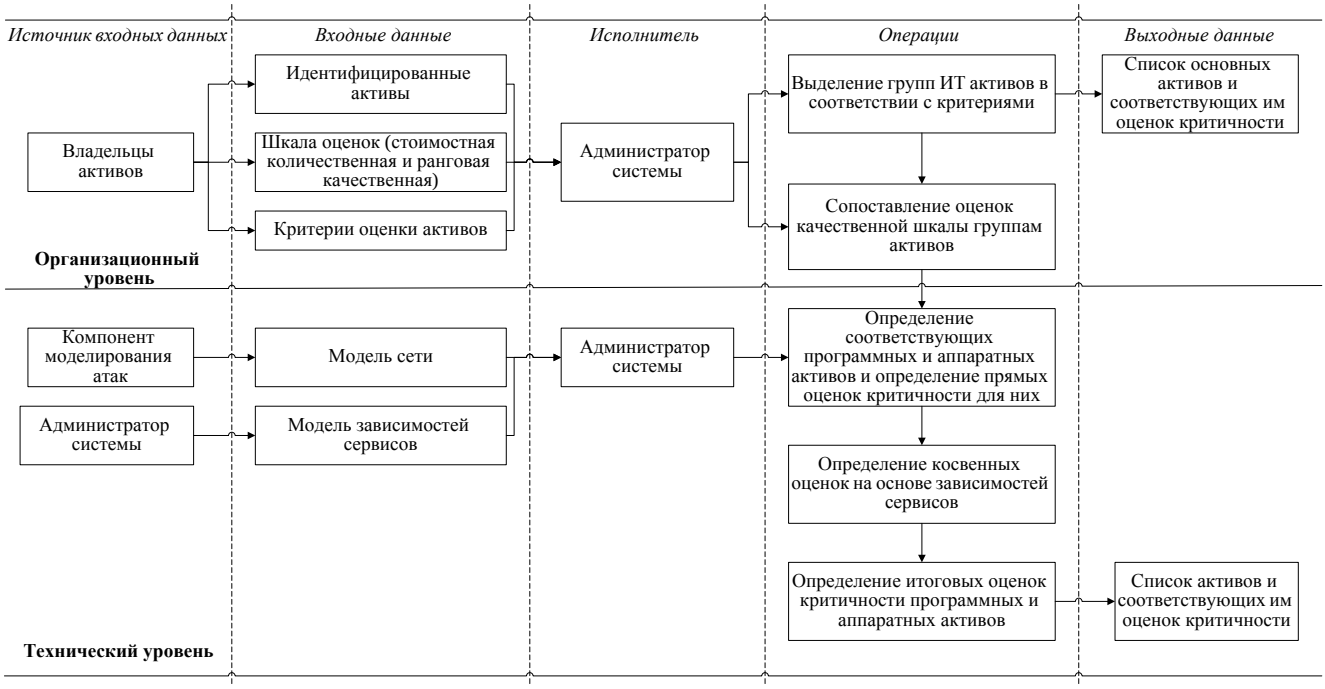


Рисунок 12 – Обобщенная схема алгоритма оценки критичности активов ИС

На первом этапе, организационного уровня, определяются ИТ активы, т.е. информация и программно-аппаратное обеспечение, непосредственно необходимое для поддержания основных бизнес-активов и процессов организации, важных для ее деловой деятельности. ИТ активы распределяются по группам в соответствии с критериями оценки. В зависимости от группы (критерия) и степени вовлеченности актива в выполнение миссии организации, им в соответствие ставятся качественные оценки, определяемые на основе стоимостной ценности, соответствующей каждому критерию. Входные данные первого этапа работы алгоритма: идентифицированные ИТ активы (определяются владельцами активов), критерии оценки активов, шкала оценок.

Согласно [10] определение ценности активов осуществляется на основе восстановительной стоимости актива и последствий для бизнеса от потери или компрометации актива. Критерии последствий для бизнеса от потери или компрометации актива (нарушение конфиденциальности, целостности и доступности) должны разрабатываться с учетом уровня классификации актива, нарушения ИБ, нарушения оперативной деятельности, потери ценности бизнеса и финансовой ценности, нарушения планов и конечных сроков, ущерба для репутации, нарушения требований. В работе не стоит цели детального определения критериев оценки активов и шкалы оценки. Критерии оценки активов могут определяться, например, на основе [10]. В соответствие критериям ставится количественная шкала оценок, отражающая финансовые потери в случае потери конфиденциальности, целостности и доступности активов (то есть прямые потери в случае нарушения критерия и последующие затраты на восстановление). Для этого может использоваться шкала, предложенная в [45] для правительственных организаций, и одобренная для организаций, использующих КС, в рамках проекта MASSIF [105] (таблица 6). Например, реализация угрозы нарушения конфиденциальности личных данных (активом в данном случае выступают данные), ведет к негативному воздействию на репутацию компании, и может потребовать значительных затрат на ее восстановление, что соответствует уровню критичности «Значительная» (€10,000) в таблице 6. В рамках разработанного алгоритма количественной шкале ставится в соответствие качественная, на основе которой определяются оценки активов от 0 до 100 (чтобы сохранить отношение между различными уровнями критичности). Для данной шкалы потери в случае реализации угрозы для каждого следующего уровня сравнимы с потерями, соответствующими

десяти случаям реализации угрозы предыдущего уровня (что необходимо учитывать при формировании шкалы). Стоимостная количественная шкала может отличаться для разных организаций (сохраняя диапазон от нулевых затрат до годового бюджета организации), однако качественная шкала остается неизменной.

Таблица 6 – Шкалы оценки критичности организационных активов

Критичность	**Описание**	**Стоимость (количествен-ная шкала)**	**Ранги (качественная шкала)**
Ничтожно малая	Практически полное отсутствие ущерба в случае реализации угрозы, не требуется никаких дополнительных затрат на восстановление	€0	0
Малая	Небольшой ущерб для ценности актива, не требуется почти никаких дополнительных затрат на восстановление	€1000	0,01
Значительная	Ощутимый ущерб, хотя и небольшой, требует некоторых затрат на восстановление	€10000	0,1
Повреждающая	Ущерб для репутации и/или ресурсов организации, требует значительных затрат на восстановление	€100000	1
Серьезная	Выход из строя системы и/или потеря клиентов или партнеров по бизнесу, затраты равные стоимости полного восстановления ресурсов	€1000000	10
Смертельная	Полная компрометация и уничтожение организации, для восстановления требуется годовой бюджет организации	€10000000	100

Операции первого этапа работы алгоритма: владельцами активов основные ИТ активы $rd_i(p) \in \boldsymbol{Rd}$ делятся на группы в соответствии с критериями оценки $GrCr_j \in \boldsymbol{GrCr}$ по параметрам конфиденциальности, целостности и доступности $p=\{c,i,a\}$: $rd_i(p) \subset GrCr_j$, где $i \in [1,n]$, $j \in [1,m]$, n – количество активов, m – количество различных критериев. Оценки прямой критичности активов определяются в соответствии с группой по параметрам конфиденциальности, целостности и доступности, на шкале от 0 до 100.

Выходные данные первого этапа работы алгоритма: список основных ИТ активов организации и соответствующих им прямых оценок критичности по параметрам конфиденциальности, целостности и доступности, на шкале от 0 до 100: <*Criticality*(*c*), *Criticality*(*i*), *Criticality*(*a*)>. Например, актив «данные на сервере баз данных», критичность <10, 10, 10>.

На втором этапе, технического уровня, определяются программно-аппаратные активы КС, необходимые для поддержания основных активов организации и прямые и внешние оценки их критичности. Входные данные второго этапа работы алгоритма: модель сети, модель зависимостей сервисов, и результаты работы первого этапа.

В данной работе используется модель сети, предложенная в [43], которая включает список моделей хостов (определяемых списком программного и аппаратного обеспечения и политиками), список связей между хостами, и тип зависимости хостов. Для целей диссертационной работы в модель хостов добавлен список моделей сервисов. Под сервисом будем понимать ресурс, предоставляющий возможность выполнения задач, формирующих необходимую функциональность с точки зрения поставщиков и потребителей услуг [142]. Модель сервиса определим следующим образом: $R=(T,Cr)$, где T – тип сервиса (*ИТ активы, порт или программно-аппаратное обеспечение*). Cr – критичность сервиса; $Cr=[Cr_r(c)\ \ Cr_r(i)\ \ Cr_r(a)]$, где $Cr_r(c)$, $Cr_r(i)$, $Cr_r(a)$ – критичность сохранения свойств конфиденциальности, целостности и доступности сервиса r, соответственно.

Модель зависимостей сервисов задается следующим образом: $SG=(R, L, \varepsilon)$, где R – множество узлов графа зависимостей сервисов (сервисов), L – множество связей ($L \subseteq R \times R$), ε – множество кортежей, определяющих тип зависимости между сервисами, вида $<L_k, d_k>$, где $L_k \in L$, $d_k \in \{$И, ИЛИ$\}$. Связь определяется как: $L=(r_i, r_j, W)$, где $r_i, r_j \in R$; $r_j \in Det(r_i)$; $Det(r_i)$ – множество всех прямых потомков сервиса r_i (то есть сервисов, от свойств безопасности которых напрямую зависят свойства безопасности r_i); W – весовая матрица, определяющая степень зависимости свойств безопасности сервиса предка от свойств безопасности сервиса потомка. В [90] авторы выделяют структурные зависимости (между сервисами различных уровней модели ISO/OSI) и функциональные (между разными сервисами одного уровня). Пример структурных зависимостей: зависимость между веб-приложением, сервером JBoss и портом tcp/443 на рисунке 13. Пример функциональных зависимостей: зависимость между веб-приложением и аутентификацией на рисунке 13.

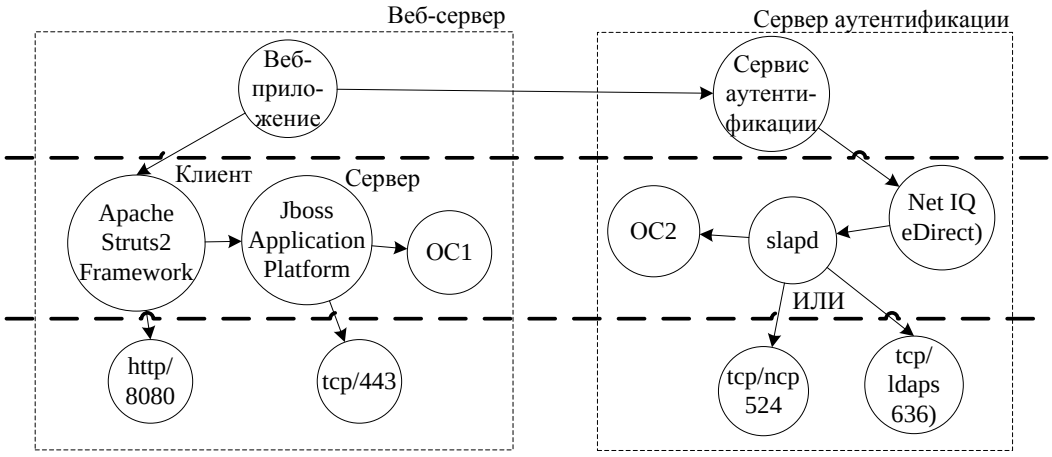


Рисунок 13 – Пример функциональных и структурных зависимостей [90]

Данная модель была выбрана, так как в отличие от других работ на эту тему, в ней учитывается не только распространение ущерба доступности, но и распространение ущерба целостности и конфиденциальности. В работе [90] модель зависимостей сервисов применяется для определения ущерба, распространяемого через зависимости сервисов, в динамическом режиме работы системы. В данной работе модель зависимостей сервисов предлагается использовать для определения показателя критичности активов в статическом режиме работы. Это связано с тем, что для определения распространения атаки в системе используется граф атак и необходимо связать распространение ущерба через зависимости сервисов с узлами графа атак (это позволит учитывать как распространение атаки в системе, так и распространение ущерба). Кроме того, это позволит экономить время в динамическом режиме работы.

Операции второго этапа работы алгоритма: выделение сервисов системы $r_k(p) \in \boldsymbol{R}$ (где $k \in [1, l]$, l – количество всех сервисов системы), необходимых для поддержания основных ИТ активов организации $rd_i(p)$; определение их прямых оценок критичности по параметрам конфиденциальности, целостности и доступности $< I\_Criticality_{r_k}(c),\ I\_Criticality_{r_k}(i),\ I\_Criticality_{r_k}(a) >$ (данные оценки уже являются параметрами компонентов модели КС); определение внешних оценок критичности на основе зависимостей сервисов $< P\_Criticality_{r_k}(c),\ P\_Criticality_{r_k}(i),\ P\_Criticality_{r_k}(a) >$; определение итоговых оценок критичности $< Criticality_{r_k}(c),\ Criticality_{r_k}(i), Criticality_{r_k}(a) >$.

Прямые оценки критичности сервисов системы по параметрам конфиденциальности, целостности и доступности $< I\_Criticality_{r_k}(c),\ I\_Criticality_{r_k}(i),\ I\_Criticality_{r_k}(a) >$ назначаются вручную экспертами. Для определения внешних оценок критичности $< P\_Criticality_{r_k}(c), P\_Criticality_{r_k}(i), P\_Criticality_{r_k}(a) >$ на основе зависимостей сервисов необходимо сформировать модель зависимостей сервисов. Сервисы частично определяются при помощи средств сканирования сети: активных (таких как Nmap [117] и Nessus [115]), и пассивных (таких как Wireshark [144]), и частично вручную администраторами. Программно-аппаратные сервисы задаются ссылками на соответствующее программно-аппаратное обеспечение (определенное с использованием словаря CPE). Зависимости между сервисами задаются

вручную администраторами, вопрос автоматического выявления зависимостей между сервисами не рассматривается в данном исследовании.

Граф зависимостей сервисов анализируется обходом в ширину сервисов по зависимостям начиная с сервисов у которых нет предков. При этом учитываются конъюнктивные зависимости от сервисов потомков (для работоспособности сервиса необходимо корректное функционирование нескольких сервисов потомков) и дизъюнктивные зависимости от сервисов потомков (для работоспособности сервиса необходимо корректное функционирование одного из сервисов потомков).

Внешняя критичность сервиса зависит от прямой критичности сервиса r_k $< I\_Criticality_{r_k}(c), I\_Criticality_{r_k}(i), I\_Criticality_{r_k}(a) >$ и критичности зависимых сервисов (сервисов предков) $r_m \in Ant(r_k)$; $Ant(r_k)$ – множество всех прямых предков сервиса r_k (то есть сервисов, напрямую зависимых от свойств безопасности r_k). Внешняя критичность сервиса определяется следующим образом. Выделяются поддеревья всех ИТ ресурсов (то есть корней графа зависимостей сервисов, с которых начинается работа алгоритма), так как критичность каждого сервиса системы зависит именно от них (они определяют финансовые потери организации в случае нарушений защищенности). Поддерево задается дополнительным индексом в модели сервиса. Внутри одного поддерева (то есть потомки одного ИТ ресурса) критичность любого сервиса потомка по трем параметрам (конфиденциальности, целостности и доступности) не может превысить критичности корневого сервиса. Для корневого сервиса r_0 внешняя критичность равна его прямой критичности:

$$P\_Criticality_{r_0}(c) = I\_Criticality_{r_0}(c);\ P\_Criticality_{r_0}(i) = I\_Criticality_{r_0}(i);$$

$$P\_Criticality_{r_0}(a) = I\_Criticality_{r_0}(a).$$

Критичность от предка к потомкам распространяется линейно в соответствии с весовой матрицей связи. При этом к внешней критичности добавляется прямая критичность сервиса (незначительная по сравнению с критичностью основных ИТ ресурсов):

$$\left[P\_Criticality_{r_k}(c) \quad P\_Criticality_{r_k}(i) \quad P\_Criticality_{r_k}(a) \right] =$$

$$= \left[wCriticality_{r_m}(c) \quad wCriticality_{r_m}(i) \quad wCriticality_{r_m}(a) \right] +$$

$$+\left[I\_Criticality_{r_k}(c) \quad I\_Criticality_{r_k}(i) \quad I\_Criticality_{r_k}(a) \right],$$

где $wCriticality_{r_m}(c) = \max(P\_Criticality_{r_m}(c) \cdot w_{r_m r_k}[c,c],$

$$P\_Criticality_{r_m}(i) \cdot w_{r_m r_k}[i,c], P\_Criticality_{r_m}(a) \cdot w_{r_m r_k}[a,c]);$$

$wCriticality_{r_m}(i) = \max(P\_Criticality_{r_m}(\mathrm{c}) \cdot w_{r_m r_k}[c,i],$

$$P\_Criticality_{r_m}(i) \cdot w_{r_m r_k}[i,i], P\_Criticality_{r_m}(a) \cdot w_{r_m r_k}[a,i]);$$

$wCriticality_{r_m}(a) = \max(P\_Criticality_{r_m}(\mathrm{c}) \cdot w_{r_m r_k}[c,a],$

$$P\_Criticality_{r_m}(i) \cdot w_{r_m r_k}[i,a], P\_Criticality_{r_m}(a) \cdot w_{r_m r_k}[a,a]);$$

$w_{r_m r_k}$ – элементы матрицы $W_{r_m r_k} = \begin{pmatrix} w_{r_m r_k}[c,c] & w_{r_m r_k}[c,i] & w_{r_m r_k}[c,a] \\ w_{r_m r_k}[i,c] & w_{r_m r_k}[i,i] & w_{r_m r_k}[i,a] \\ w_{r_m r_k}[a,c] & w_{r_m r_k}[a,i] & w_{r_m r_k}[a,a] \end{pmatrix}$.

Если итоговая внешняя критичность превышает максимальное значение шкалы критичности, то она приравнивается к максимальной оценке.

Если у сервиса несколько сервисов предков, то внутри одного поддерева критичность сервиса определяется максимальной критичностью сервисов предков:

$$\left[P\_Criticality_{r_k}(c) \quad P\_Criticality_{r_k}(i) \quad P\_Criticality_{r_k}(a) \right] =$$

$$= \left[\max_m P\_Criticality_{r_k}(c) \quad \max_m P\_Criticality_{r_k}(i) \quad \max_m P\_Criticality_{r_k}(a) \right],$$

где $m \in [1, S]$; S – количество сервисов предков.

Если сервис потомок имеет предков из разных поддеревьев, то его итоговая критичность определяется суммой критичности сервисов предков, но не может превышать максимальный уровень критичности (100):

$$\left[P\_Criticality_{r_k}(c) \quad P\_Criticality_{r_k}(i) \quad P\_Criticality_{r_k}(a) \right] =$$

$$= \sum_m \begin{pmatrix} \left[P\_Criticality_{r_m}(c) \quad P\_Criticality_{r_m}(i) \quad P\_Criticality_{r_m}(a) \right] \cdot W_{r_m r_k} + \\ + \left[I\_Criticality_{r_m}(c) \quad I\_Criticality_{r_m}(i) \quad I\_Criticality_{r_m}(a) \right] \end{pmatrix},$$

где m – сервисы предки из разных поддеревьев.

Если у сервиса предка несколько сервисов потомков, то распространение критичности зависит от типа зависимости: для конъюнктивных зависимостей

критичность распространяется линейно в соответствии с формулой выше; для дизъюнктивных зависимостей критичность делится между сервисами потомками:

$$\left[P\_Criticality_{r_k}(c) \quad P\_Criticality_{r_k}(i) \quad P\_Criticality_{r_k}(a) \right] =$$

$$= \left[\frac{P\_Criticality_{r_k}(c)}{\sum_{k=1}^{K} P\_Criticality_{r_k}(c)} \quad \frac{P\_Criticality_{r_k}(i)}{\sum_{k=1}^{K} P\_Criticality_{r_k}(i)} \quad \frac{P\_Criticality_{r_k}(a)}{\sum_{k=1}^{K} P\_Criticality_{r_k}(a)} \right] +$$

$$+ \left[I\_Criticality_{r_k}(c) \quad I\_Criticality_{r_k}(i) \quad I\_Criticality_{r_k}(a) \right],$$

где K – количество дизъюнктивных связей.

Если итоговая внешняя критичность превышает максимальное значение шкалы критичности, то она приравнивается к максимальной оценке.

Выходные данные второго этапа работы алгоритма: список сервисов и соответствующих им оценок критичности.

Преимущества алгоритма: учет не только основных активов, непосредственно участвующих в деятельности организации, но и активов, необходимых для их корректного функционирования, что позволяет с одной стороны, учесть косвенный ущерб, с другой стороны, избежать побочного ущерба при реализации контрмер; прямая зависимость между стоимостью активов для организации и шкалой оценки критичности, позволяющая обосновать необходимость защиты активов. Ограничения алгоритма: при распространении критичности сервиса критичность для сервиса потомка рассчитывается на основе максимальной распространенной критичности сервиса предка по параметрам конфиденциальности, целостности и доступности. В случае если сервис критичен для всех трех свойств безопасности, его критичность будет занижена. Учет распространения критичности по всем трем параметрам требует применения рекурсивного алгоритма, обладающего высокой вычислительной сложностью. Поэтому для сохранения ресурсов времени и памяти данный алгоритм упрощен.

Алгоритм оценки ущерба от атакующего действия. Ущерб от атакующего действия определяется на основе критичности актива и разрушительности атакующего действия. Разрушительность атакующего действия определяется на основе базовых показателей CVSS для уязвимостей [109]: влияние на конфиденциальность, целостность и доступность. Данный показатель определяет влияние на свойства безопасности актива R_k ($k \in [1, l]$, l – количество всех программных активов организации) атакующего

действия a_i в результате успешной эксплуатации уязвимости v_i ($i \in [1,m]$, m – множество всех уязвимостей данного актива), в виде трехзначного вектора $\left[ConfImpact_{k,i}(c) \quad IntegImpact_{k,i}(i) \quad AvailImpact_{k,i}(a) \right]$, где $ConfImpact_{k,i}(c)$ – влияние на конфиденциальность актива R_k, $IntegImpact_{k,i}(c)$ – влияние на целостность актива, $AvailImpact_{k,i}(c)$ – влияние на доступность актива. $ConfImpact_{k,i}(c)$, $IntegImpact_{k,i}(c)$ и $AvailImpact_{k,i}(c)$ могут принимать значения {0; 0,275; 0,66} в соответствии с возможными значениями показателей CVSS *влияние на конфиденциальность*, *влияние на целостность* и *влияние на доступность*.

Ущерб от атакующего действия $\left[Impact_{k,i}(c) \quad Impact_{k,i}(i) \quad Impact_{k,i}(a) \right]$ определяется по свойствам конфиденциальности $Impact_{k,i}(c)$, целостности $Impact_{k,i}(i)$ и доступности $Impact_{k,i}(a)$ путем перемножения показателей *критичности актива* по соответствующему свойству и *разрушительности атакующего действия*. Общий ущерб определяется суммированием ущерба по трем свойствам.

Алгоритм оценки вероятности атакующего действия. Для определения *вероятности атакующего действия* a_i, использующего уязвимость v_i, используется показатель CVSS *Exploitability*: $Exploitability = 20 \times AccessVector \times AccessComplexity \times$ $\times Authentication$, где $AccessVector$ – *вектор доступа*; $AccessComplexity$ – *сложность доступа*; $Authentication$ – *аутентификация*.

Алгоритмы вычисления показателей уровня графа атак. Модель атак, используемая для вычисления показателей в данной работе, является развитием моделей, предложенных в диссертационных исследованиях [37, 43]. Она задается следующим образом: $G=(S, L, Pc)$, где S – множество узлов графа (атакующих действий), L – множество связей ($L \subseteq S \times S$); Pc – дискретные локальные распределения условных вероятностей.

Модель атакующих действий определяется как $S=(H, V, Sc, St, Pr)$, где H – определяет атакованный хост (включает описание сервисов хоста), V – использованную уязвимость, Sc – атаку, направленную на сбор информации о хосте, St – состояние атакующего действия ($St=\{True, False\}$), Pr – вероятность того, что атакующее действие находится в состоянии St ($Pr \in [0,1]$). Модель уязвимости V задается как $V = (P, R, I)$, где R и I определяют результат атакующего действия (получение прав доступа и/или воздействие на информацию), а P – определяет необходимые условия для возможности

выполнения атакующего действия. Условия применимости уязвимости P определяются как $P = (AT, R)$, где AT – необходимый тип доступа ($AT = \{remote, local\}$), а R – необходимые привилегии. Состояние атакующего действия St вводится для последующего учета динамического характера атак. Под атакой понимается последовательность атакующих действий (узлов графа), реализующих угрозу для некоторого актива.

Узел графа является случайной переменной Бернулли, отображающей состояние атакующего действия. Такой тип графов атак относится к Байесовским графам атак. Данная модель была выбрана, так как байесовские графы атак позволяют учитывать влияние событий на состояние системы и прогнозировать в соответствии с этим развитие атаки и предыдущие шаги атаки. Кроме того, они позволяют делать выводы об атаке на основе субъективных знаний при отсутствии статистических данных об успешном использовании уязвимостей сети.

Узел графа атак считается скомпрометированным ($St=True$), если уязвимость V успешно проэксплуатирована. Уязвимость V входит в группу уязвимостей, соответствующую данному узлу графа, включающую все уязвимости, которые позволяют выполнить соответствующее атакующее действие. Уязвимости сгруппированы в соответствии с индексами CVSS, определяющими предусловия (AV – вектор доступа к уязвимости) и постусловия ($priv$ – полученные привилегии и/или CIA – нанесенный ущерб) их эксплуатации. Это позволяет сформировать связи между узлами графа атак, а также снизить размерность графа. Выделяются группы уязвимостей:

- группа 1: AV=N/A (N – сетевой доступ, A – доступ из смежной сети), $priv$=user/other (привилегии пользователя или другие), CIA=any (любой);

- группа 2: AV=N/A, $priv$=admin (администраторские привилегии), CIA=any;

- группа 3: AV=N/A, $priv$=none (не дает привилегий), CIA=P/C (частичный или полный ущерб);

- группа 4: AV=L (локальный доступ), $priv$=admin, CIA=any;

- группа 5: AV=L, $priv$=user/other, $CIA > CIA_{группа1}$ (остальные уязвимости отсекаются, так как их эксплуатация не имеет смысла);

- группа 6: AV=L, $priv$=none, $CIA > CIA_{группа1}$.

На рисунке 14 представлены связи между атакующими действиями, использующими уязвимости соответствующей группы в рамках одного хоста.

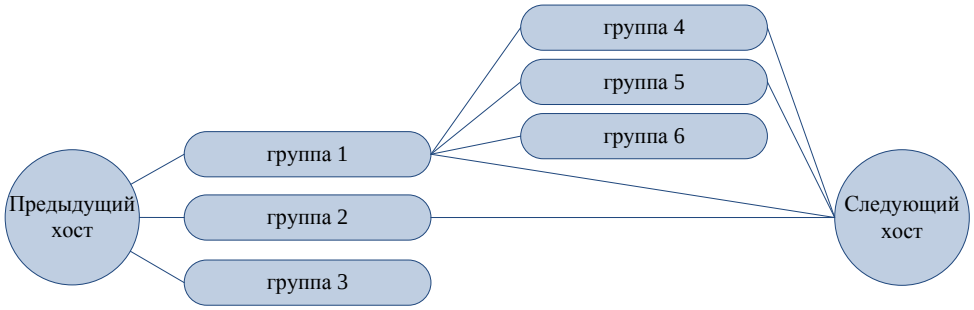


Рисунок 14 – Связи между группами уязвимостей

В статическом режиме все узлы находятся в нескомпрометированном состоянии и имеют вероятности перехода в скомпрометированное состояние. Связи определяют переходы между узлами графа. Для перехода в новое состояние необходимо наличие связи и успешная эксплуатация уязвимости узла. Для графа определены два типа отношений между связями: И – для перехода в скомпрометированное состояние необходимо скомпрометировать все узлы-предки, связанные данным отношением (цепочка последовательно связанных узлов графа); ИЛИ – необходимо скомпрометировать хотя бы один из узлов-предков, связанных данным отношением (узлы графа, находящиеся на одном уровне).

Ограничения модели: предполагается, что граф обладает свойством монотонности (то есть атакующий не возвращается назад и движется в направлении увеличения выигрыша) [81, 129]; граф не имеет циклов, так как для атакующего не имеет смысла повторное посещение уже посещенных узлов [66, 125] (это ведет к завышению оценок вероятности атаки); события компрометации узлов графа предполагаются независимыми.

Вычисление вспомогательных показателей уровня графа атак. В качестве вспомогательных были выбраны показатели: *нормализованное количество атак, проходящих через хост*, *нормализованное количество атак с высоким уровнем риска, проходящих через хост*. Показатели измеряются в процентах на шкале от 0 до 100 и позволяют выделить хосты, через которые проходит наибольшее количество атак (слабые места КС).

Показатель *нормализованное количество атак, проходящих через хост QNAH* предлагается вычислять по формуле: $QNAH = \frac{n_i}{n}$, где n_i – количество атак, проходящих через хост i; n – общее количество атак графа.

Показатель *нормализованное количество атак с высоким уровнем риска, проходящих через хост HQNAH* предлагается вычислять по формуле: $HQNAH = \frac{nh_i}{nh}$,

где nh_i – количество атак с высоким уровнем риска, проходящих через хост i; nh – общее количество атак графа с высоким уровнем риска.

Алгоритмы вычисления основных показателей уровня графа атак. Для вычисления основных показателей уровня графа атак применяется модель атак (в форме Байесовского графа атак) и характеристики уязвимостей, применяемых при реализации атакующих действий, полученные из базы NVD [119]. В качестве основных были выбраны показатели: *сложность атакующего действия*; *сложность атаки*; *вероятность атаки*; *разрушительность атакующего действия*; *ущерб от атаки*.

Показатель *сложность атаки* вычисляется на основе индекса CVSS *сложность доступа* [109]. Пусть *AccessComplexity*(v_i) – *сложность доступа* уязвимости v_i, применяемой для реализации атакующего действия a_i. Тогда *сложность атакующего действия* a_i равна *AccessComplexity*(v_i). Сложность атаки *attackComplexity* определяется как max_i*AccessComplexity*(v_i), где $i \in [1, N]$, N – количество атакующих действий в атаке.

Показатель *ущерб от атаки aImpact* вычисляется на основе показателя *ущерб от атакующего действия* предыдущего уровня. Пусть *Impact*(a_i) – ущерб, наносимый КС в результате успешного выполнения атакующего действия a_i, $a_i \in A$, $i \in [1, N]$, где A – множество всех шагов атаки, N – количество шагов атаки. Тогда $aImpact = \max_i Impact(a_i)$.

Предлагаемый в данном исследовании алгоритм определения *вероятности атаки* использует и развивает предыдущие работы, применяющие байесовские графы атак [67, 70, 99, 125, 141]. Отличия: метод формирования графа атак; метод вычисления локальных вероятностей компрометации узлов.

В работах [67, 70, 99, 125, 141] предлагается применение следующего механизма для определения вероятности атаки на узел графа (байесовский метод определения вероятности события, при этом под вероятностью понимается степень доверия): (1) определение локальных вероятностей компрометации узлов графа (то есть вероятностей того, что атакующий сможет и проэксплуатирует уязвимость, соответствующую данному узлу, если все предусловия выполнены); (2) определение условных вероятностей компрометации узлов графа (дискретных локальных распределений условных вероятностей); (3) определение безусловных вероятностей компрометации узлов путем обхода графа.

Локальные вероятности компрометации узлов определим на основе индексов CVSS [109]. В [125] применяется показатель CVSS *Exploitability*. Данный показатель вычисляется, в том числе, на основе индекса CVSS *AccessVector* (*вектор доступа* к

уязвимости). Предлагаемый в данном исследовании граф атак построен таким образом, что переход из состояния в состояние возможен только в случае наличия доступа к соответствующему узлу. Поэтому при определении локальных вероятностей успешной компрометации узла данный индекс учитывается только для корневых (входных) узлов графа. Таким образом, локальные вероятности успешной компрометации узла S_i, соответствующего атакующему действию a_i, определим как: $p(a_i) = 2 \times AccessVector \times$ $\times AccessComplexity \times Authentication$, если $S_i \in S_r$, где S_r – множество корневых (входных) узлов графа, *AccessComplexity – сложность доступа* к уязвимости по CVSS; *Authentication* – требуемая *аутентификация* по CVSS. В этом случае локальная вероятность успешной компрометации узла может принимать значения от 0,1 до 1 (в соответствии с возможными значениями индексов CVSS). Если $S_i \notin S_r$: $p(a_i) = 2 \times$ $\times AccessComplexity \times Authentication$, в этом случае локальная вероятность успешной компрометации узла может принимать значения от 0,3 до 1. Вероятность того, что узел не будет скомпрометирован, определяется как 1-$p(a_i)$. Схема алгоритма определения локальных вероятностей приведена в приложении А.

Для определения дискретных локальных распределений условных вероятностей *Pc* (то есть вероятностей компрометации узла с учетом различных комбинаций состояний его предков) необходимо учесть типы связей между узлами предками. Обозначим $Pa(S_i)$ множество всех предков узла S_i, а функцию локального распределения условной вероятности $Pc(S_i|Pa(S_i))$. В работе [70] были предложены следующие две формулы. Для определения условной вероятности в случае связей типа «И» между узлами предками (для успешной компрометации узла потомка необходимо, чтобы все узлы предки были скомпрометированы): $Pc(S_i \mid Pa(S_i)) = \begin{cases} 0, & \exists S_i \in Pa(S_i) \mid S_i = 0 \\ p(S_i), & \text{иначе} \end{cases}$.

Для определения условной вероятности в случае связей типа «ИЛИ» между узлами предками (для успешной компрометации узла потомка необходимо, чтобы хотя бы один узел предок был скомпрометирован): $Pc(S_i \mid Pa(S_i)) = \begin{cases} 0, & \forall S_i \in Pa(S_i) \mid S_i = 0 \\ p(S_i), & \text{иначе} \end{cases}$.

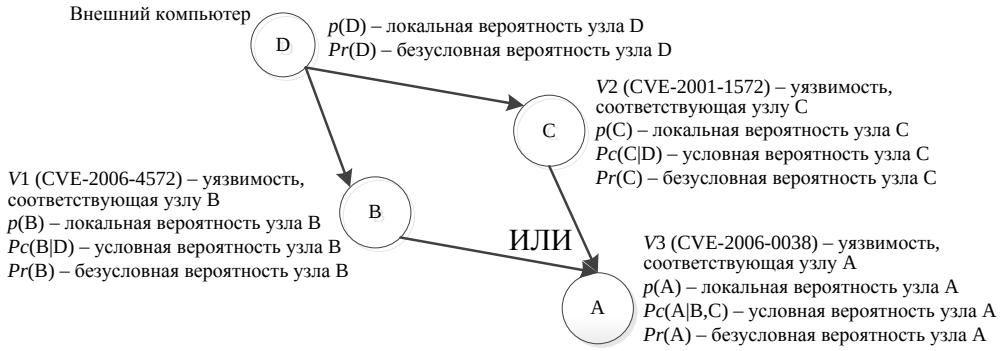
Для определения условных распределений вероятностей всех узлов применяется обратный обход графа в глубину, начиная с терминальных узлов (не имеющих потомков) и заканчивая узлами, доступными атакующему. Схемы алгоритмов определения условных вероятностей приведены в приложении Б.

Безусловные вероятности компрометации узлов графа (*вероятности атаки*) определяются на основе локальных вероятностей и распределений условных

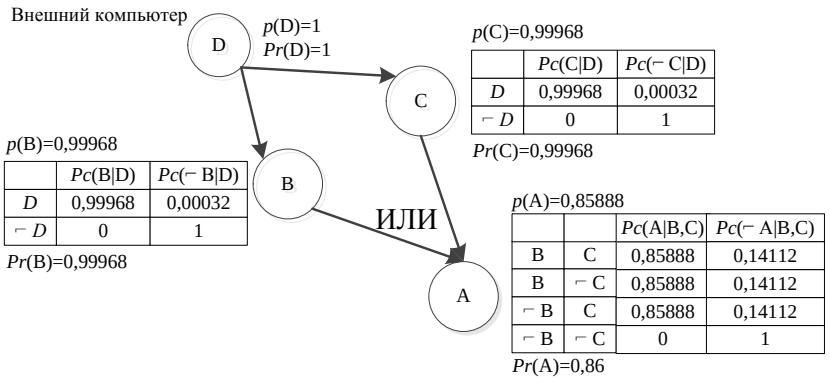
вероятностей по формуле полной вероятности путем маргинализации по известным вероятностям: $Pr(S_1,...,S_n)=\prod_{i=1}^{n} Pc(S_i \mid Pa[S_i])$, где S_i – i-й узел графа. Схема алгоритма определения безусловных вероятностей приведена в приложении В.

На рисунке 15, а представлен фрагмент графа атак, каждому узлу графа сопоставлена уязвимость и набор показателей (локальная вероятность, условная вероятность и безусловная вероятность). На рисунке 15, б для этого фрагмента приведены численные значения, рассчитанные в соответствии с формулами выше. Подробно процесс вычисления вероятностей для графа на рисунке 15 приведен в приложении Г.

Выходные данные работы алгоритма: итоговые значения *вероятностей атаки* для всех узлов графа.



а)



б)

Рисунок 15 – Фрагмент байесовского графа атак

Алгоритмы вычисления показателей уровня атакующего. Для определения показателей уровня атакующего применяется модель атакующего, которая задается как [43]: $A = (H_0, Sk, G)$, где H_0 ($H_0 \subseteq H$) – определяет хосты, к которым имеет доступ атакующий до проведения атак и уровень привилегий на них, G – содержит цели

атакующего ($G = <H \times I, H \times R>$, где R и I определяют результат атакующего действия: получение прав доступа и/или воздействие на информацию), Sk – определяет уровень навыков атакующего.

Показатель *уровень навыков атакующего* задается администратором на шкале: Sk = {None, Low, Medium, High}. Данным качественным оценкам ставятся в соответствие количественные: Sk = {0, 0,35, 0,61, 0,71} (по аналогии с показателем *сложность доступа* CVSS).

Профильная вероятность атаки определяется следующим образом: (1) на основе списка хостов, к которым имеет доступ атакующий до проведения атак, и уровней привилегий на них, формируется профильный граф атак, то есть подграф, к которому имеет доступ атакующий; (2) локальные вероятности переопределяются по формуле: $p=AV\times(AC+Sk)\times Au$, для корневых узлов графа, и $p = \frac{(AC + Sk)}{2} \times Au$, для остальных узлов, где AV – *вектор доступа* к уязвимости по CVSS; AC – *сложность доступа* к уязвимости по CVSS; Au – *аутентификация* по CVSS; (3) далее применяется алгоритм определения *вероятности атаки* уровня графа атак.

Алгоритмы вычисления показателей уровня событий. Под событием безопасности будем понимать обработанное событие, поступающее от SIEM-системы. Например: получение нелегитимных привилегий на хосте; нарушение конфиденциальности хоста. В рамках проекта MASSIF [105] события безопасности, прежде чем поступить на вход компонента оценки защищенности, проходят через систему корреляции и обработки событий (рисунок 5), которая извлекает так называемое «сырое» событие (т.е. текстовое сообщение, содержащее все данные исходного сообщения) из сообщения, формирует сообщение необходимого формата на основе извлеченных данных, и на основе правил корреляции формирует события более высокого уровня.

На уровне событий состояние графа атак S_i меняется ввиду поступления нового события ev_i, определяющего, что один из узлов графа атак перешел в состояние «скомпрометирован». Подобная формализация предложена в [91] для моделирования распространения ущерба в графах зависимостей сервисов при поступлении внешнего события об атаке от системы обнаружения вторжений (СОВ). Новое состояние графа S_{i+1} характеризуется тем, что оценка вероятности компрометации соответствующего узла изменилась (рисунок 16). В результате меняются оценки вероятности компрометации и рисков для последующих узлов графа атак.

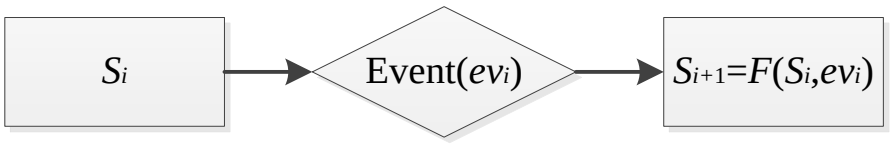


Рисунок 16 – Изменение состояния системы в результате поступления события

Согласно классификации [78] для описания события безопасности используются категории: атакующий (кто проводит атаку); метод атаки (методы эксплуатации уязвимости); уязвимость (тип использованной уязвимости); действие (шаги для достижения результата); объект атаки (на что направлена атака); результат атаки (последствия инцидента); цели (цели атаки). В исследовании для отображения события безопасности на граф атакующих действий используются поля: объект атаки (учетная запись, процесс, данные, компонент, компьютер, сеть); результат атаки (эскалация привилегий, раскрытие информации, разрушение информации, отказ в обслуживании, кража ресурсов). При наличии, можно учитывать дополнительную информацию: атакующий (кто проводит атаку); цели (цели атаки). Модель события задается следующим образом: $E = (Ti, H, Te)$, где Ti — время произошедшего изменения (в данном исследовании рассматриваются дискретные моменты времени); H — модель хоста, на котором произошло событие; Te — тип события ($Te = \{R, I,$ other$\}$, где R — получение прав доступа, I — воздействие на информацию, т.е. нарушение конфиденциальности/целостности/доступности).

Вычисление вспомогательных показателей уровня событий. Показатель *нормированное количество инцидентов на хосте* отображает количество инцидентов на хосте по отношению к общему количеству инцидентов в системе: $NNHI = \frac{NI_i}{NI}$, где NI_i – количество событий безопасности, поступивших от SIEM-системы для i-го хоста (*количество инцидентов на хосте*); NI – общее количество событий безопасности, поступиших от SIEM-системы по всем хостам сети (*количество инцидентов в системе*). Показатель измеряется в процентах, и может изменяться во времени.

Показатель *надежность события* определяется на основе показателя надежности, поступающего с событием безопасности от SIEM-системы.

Алгоритмы вычисления основных показателей уровня событий. На данном уровне вычисляются значения следующих основных показателей: *динамический уровень навыков атакующего*; *динамическая вероятность атаки*.

Входные данные вычисления показателей: входные данные и показатели предыдущих уровней; модель события.

Алгоритм вычисления основных показателей уровня событий включает этапы: (1) отображение события на граф атак; (2) переопределение уровня навыков атакующего; (3) переопределение вероятности атаки. Схема алгоритма отображения события безопасности на граф атакующих действий представлена на рисунке 17. Алгоритм заключается в поиске узла графа по хосту, которому соответствует событие безопасности и последствиям атакующего действия, сгенерировавшего событие. Результатом работы алгоритма является позиция атакующего на графе атак.

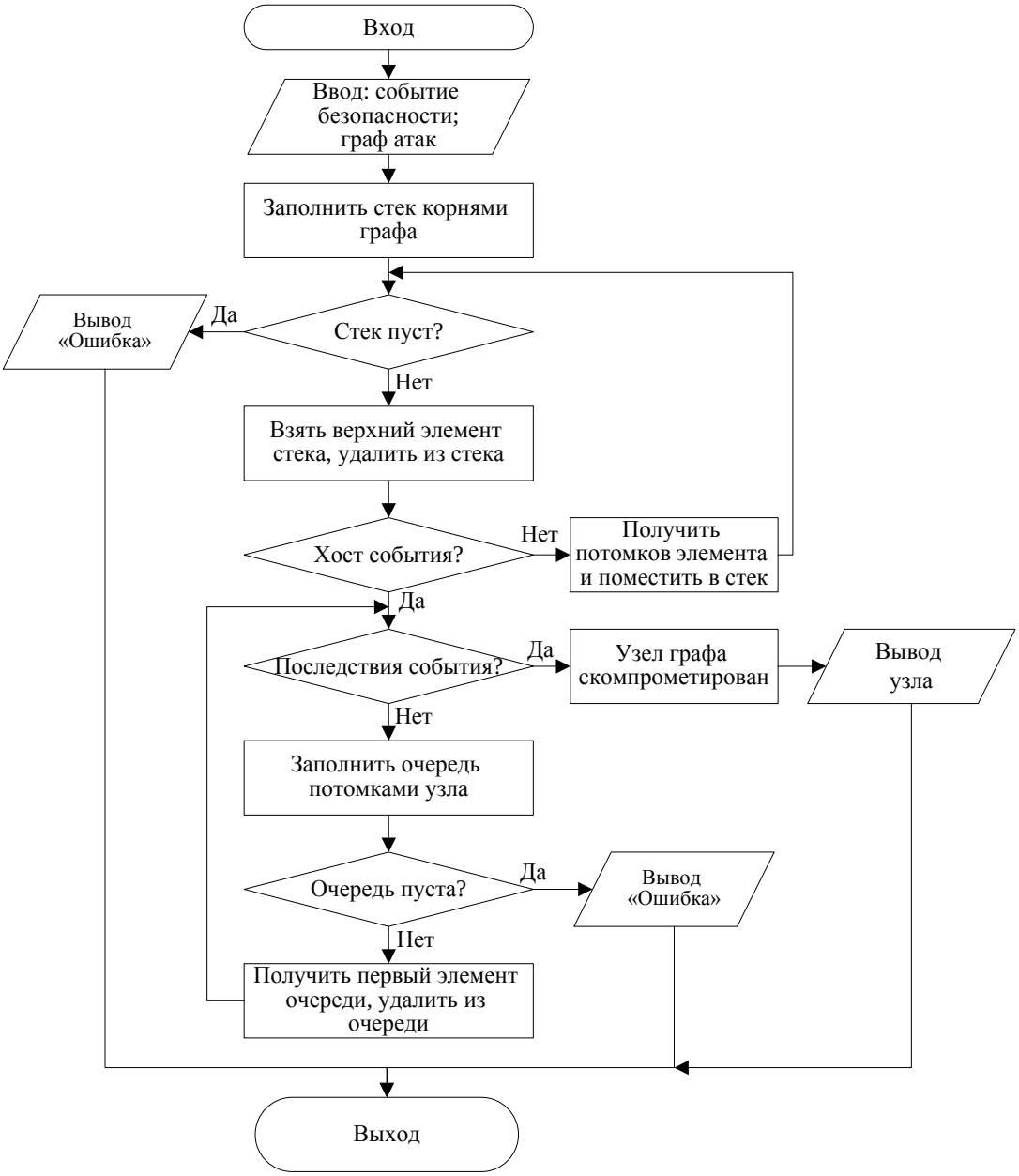


Рисунок 17 – Алгоритм отображения события безопасности на граф атакующих действий

Динамический уровень навыков атакующего определяется на основе шагов:

1) Перевод узла, соответствующего позиции атакующего на графе в состояние «скомпрометирован».

2) Переопределение вероятности атаки для узла, для которого поступило событие безопасности (на основе теоремы Байеса). Вместе с событием безопасности от SIEM-системы поступает значение *надежности информации*, которое определяет вероятность того, что SIEM-система сообщит об атаке, если она произошла $p(ev|a)$ и значение вероятности того, что атака не произошла, если SIEM-система сообщает об этом (false positive) $p(ev|\neg a)$. Тогда вероятность того, что узел скомпрометирован $p(a|ev)$: $p(a|ev) = \frac{p(ev|a) \times p(ev)}{p(a)} = \frac{p(ev|a)}{p(a)} \times (p(ev|a) \times p(a) + p(ev|\neg a) \times p(\neg a))$, где $p(a)$ — вероятность компрометации узла (определенная для узла ранее на основе графа атак); $p(ev)$ – безусловная вероятность получения события от SIEM-системы.

3) Определение наиболее вероятного пути атакующего до узла, выбранного на предыдущем шаге (на основе теоремы Байеса). То есть для всех предков узла, выбранного на предыдущем шаге, определение вероятности того, что был атакован узел предок b при условии компрометации узла потомка a: $p(b|a) = \frac{p(a|b) \times \mathrm{p}(b)}{p(a)}$, где $p(a|b)$ — вероятность компрометации узла потомка при условии компрометации узла предка (условная вероятность, которая была определена на основе графа атак); $p(b)$ — полная вероятность компрометации узла предка (безусловная вероятность, которая была определена на основе графа атак); $p(a)$ — вероятность компрометации узла потомка, определенная на предыдущем шаге. В случае нескольких путей с одинаковыми значениями вероятности в последующих вычислениях участвуют все пути.

4) Выбор узлов пути с максимальным значением индекса CVSS *сложность доступа*. *Динамический уровень навыков атакующего* определяется равным этому значению (значение на шкале «Высокий»/«Средний»/«Низкий», чему соответствуют количественные значения: 0,7, 0,5 и 0,3).

5) Определение точности показателя как отношения узлов пути с соответствующим уровнем *сложности доступа* к общему количеству узлов пути.

Результат работы алгоритма: наиболее вероятный путь до позиции атакующего на графе атак; *динамическая вероятность атаки* для узла, соответствующего позиции

атакующего на графе, и его предков; *динамический уровень навыков атакующего* и точность определения уровня навыков атакующего.

Показатель *динамическая вероятность атаки* для потомков узла, соответствующего позиции атакующего на графе, определяется с учетом *динамической вероятности атаки* для данного узла и *уровня навыков атакующего* на основе шагов:

1) Перевод узла, соответствующего позиции атакующего на графе в состояние «скомпрометирован».

2) Переопределение вероятности атаки для узла, для которого поступило событие безопасности (шаг 2 методики определения уровня навыков атакующего).

3) Вычисление новых значений локальных вероятностей для путей атак, проходящих через данный узел с учетом *динамического уровня навыков атакующего*.

4) Вычисление вероятностей для путей атак, проходящих через данный узел по формуле полной вероятности, с учетом нового значения вероятности скомпрометированного узла и с учетом новых значений локальных вероятностей.

Результатом работы алгоритма является: *динамическая вероятность атаки*.

Алгоритмы вычисления показателей уровня выбора контрмер. Уровень защищенности в статическом режиме и распространение атаки в динамическом режиме зависит от того, реализованы или нет защитные меры M_i. Защитные меры влияют на изменение состояния графа атак при реализации атакующего действия (рисунок 18).

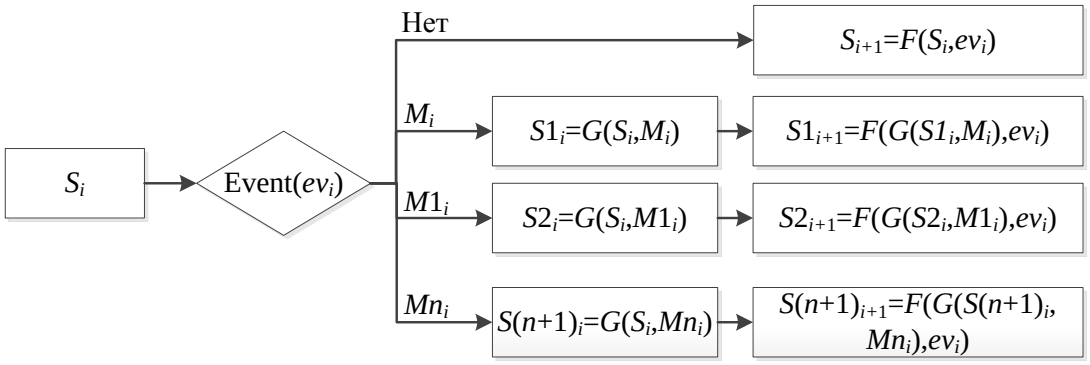


Рисунок 18 – Влияние защитных мер на состояние графа атак [91]

Для вычисления показателей уровня выбора контрмер вводится модель контрмеры. В данном исследовании для создания модели контрмер используются стандарты протокола SCAP: CRE [107] и ERI [86]. Модель контрмеры определяет: характеристики контрмеры, необходимые для ее включения в методику выбора

контрмер; возможные значения характеристик контрмеры; связь характеристик контрмеры с методикой выбора контрмер.

Предлагаемая концептуальная модель контрмеры включает поля (рисунок 19):

1) Поля, описывающие контрмеру: название контрмеры (текстовое поле); описание контрмеры (текстовое поле), поле унаследовано из стандарта CRE.

2) Поля, определяющие связь с графом атак: тип влияния на граф атак – удаление, добавление или изменение узла/связи в графе (текстовое поле, принимает значения {REMOVE<идентификатор узла/дуги>, ADD<идентификатор узла/дуги>, MODIFY<идентификатор узла/дуги>}); уязвимость или конфигурация, против которой может использоваться контрмера (ссылка на CVE или CCE), поле унаследовано из стандарта ERI; платформа для которой может использоваться контрмера (ссылка на CPE), поле унаследовано из стандарта CRE.

3) Поля, определяющие связь с методикой выбора контрмер: средство реализации (текстовое поле), заполняется экспертами; режим работы системы (может принимать значения: статический, динамический, оба); область действия (может принимать значения: элемент графа атак, хост, подсеть, сеть).

4) Показатели защищенности: *Эффективность контрмеры* (определяется экспертами); *Стоимость контрмеры* (определяется экспертами); *Уровень побочного ущерба* (определяется на основе графа зависимостей сервисов).

Поля:

описание		связь с графом атак		связь с методикой выбора контрмер				показатели		
название	**описание**	**тип влияния на граф атак**	**платформа**	**CCE или CVE**	**средство реализации**	**режим работы**	**область действия**	**побочный ущерб**	**эффективность**	**стоимость**

Пример значений полей:

Запрет или перенаправ-ление запросов	Запрет или перенаправление url запросов от подозрительных учетных записей	Удаление связи	cpe:/ a:apache:struts: 2.0.0	CVE-2010-1870	Межсетевой экран	Динамический	Подсеть	*CD* = [0 0 0,5]	*CE* = [0,5 0,5 0,5]	500 €

Рисунок 19 – Поля концептуальной модели контрмеры

В данном исследовании выделяются статический и динамический режимы работы. В статическом режиме контрмеры включают различные инструменты, которые позволяют снизить уровень риска. В динамическом режиме рассматриваются контрмеры, которые могут предотвратить распространение атаки вглубь КС. Например, в статическом режиме в систему может быть добавлен межсетевой экран, а в динамическом режиме он может применяться для блокировки подозрительных учетных

записей. Данное разделение основано на предположении, что в динамическом режиме работы системы нет времени на развертывание дополнительных инструментов защиты. Примеры контрмер согласно [10]:

- Идентификация и аутентификация. В статическом режиме для идентификации и аутентификации может использоваться программный токен. В динамическом режиме примером является активация многофакторной аутентификации с использованием программного токена.

- Логическое управление и аудит доступа. В статическом режиме может быть реализовано путем мониторинга сети с использованием продуктов NetCrunch [116] или FreeNATS [69]. В динамическом режиме данные продукты могут использоваться для активации правил аномального поведения.

- Обнаружение и предотвращение вторжений. В статическом режиме для обнаружения и предотвращения вторжений могут использоваться СОВ и системы предотвращения вторжений (СПВ). В динамическом режиме они позволяют реализовать контрмеры: активация СОВ и СПВ в стратегических местах, смена портов соединения. Примеры СОВ на хостах: Intrust [83], Snort [130]. Пример СПВ: Cisco 500 Se [58].

- Предохранение от злонамеренного кода. Для предохранения от злонамеренного кода в статическом режиме могут использоваться сканеры. В динамическом режиме они позволяют реализовать удаление злонамеренного кода. Пример антивирусного сканера: Kaspersky [88].

- Управление безопасностью сети (планирование, эксплуатация и администрирование сетей). Для обеспечения безопасности сети в статическом режиме могут использоваться межсетевые переходы безопасности, виртуальные частные сети, СОВ и СПВ, мониторинг сети, межсетевые экраны. В динамическом режиме СОВ и СПВ позволяют реализовать контрмеры: активация СОВ в стратегических местах, смена портов соединения. Мониторинг сети позволяет активировать правила аномального поведения. Межсетевые экраны позволяют реализовать контрмеры: блокировка подозрительных учетных записей, временная деактивация учетных записей пользователей (на 24, 48 или 72 часа). Примеры межсетевых экранов: Comodo [63], Endian [68]. Кроме того можно усилить защищенность сети путем обновления ПО (и удаления уязвимостей).

- Криптография. Для обеспечения конфиденциальности, целостности, неотказуемости и аутентичности, в том числе при передаче данных по сети, применяются различные криптографические алгоритмы и протоколы передачи данных.

Выбор данных контрмер обусловлен тем, что они относятся к специальным защитным мерам, применяемым в системах, для которых требуется детальный анализ рисков (рассматриваемым в данном исследовании).

В таблице 7 представлена связь контрмер с угрозами различным свойствам безопасности. В таблице применяются обозначения: S – применяется в статическом режиме, D – применяется в динамическом режиме, SD – применяется в обоих режимах; C – конфиденциальность, I – целостность, A – доступность. Классификация основана на стандарте [10].

Таблица 7 – Примеры угроз, свойств безопасности и контрмер

Примеры угроз	**Свойство безопасности**	**Примеры контрмер**							
		Предохранение от злонамеренного кода	Идентификация и аутентификация	Логическое управление и аудит доступа	Управление безопасностью сети	Криптография	Обнаружение и предотвращение вторжений	Резервные копии	Управление персоналом
Злонамеренный код	C	SD					D		
	I	SD					D	SD	
	A	SD					D		
Подмена личности пользователя	C	SD	S	SD	SD	S			
	I	SD	S	SD	SD	S		SD	
	A	SD	S	SD	SD	S		SD	
Ложная маршрутизация/перенаправление сообщения	C				SD	S			
	I				SD	S			
	A				SD	S			
Несанкционированный доступ к компьютерам, данным, сервисам и приложениям	C		S	SD	SD	S			
	I		S	SD	SD	S		SD	
	A		S	SD	SD	S			
Разрушительная атака	C								
	I								
	A		S	SD				SD	S
Неправильное использование ресурсов	C								
	I								
	A		S	SD	SD				S
Перегрузка трафика	C								
	I								
	A				SD			SD	

При создании модели учитывалось применение графа атак и графа зависимостей сервисов для оценки защищенности. Граф зависимостей сервисов применяется при определении побочного ущерба, наносимого в результате реализации контрмер. Кроме того, реализация контрмеры влияет на переходы состояний и, соответственно, изменяет

граф атак. Очевидно, что контрмера может повлиять на каждый из элементов графа атак (узел и дуга) тремя способами: удаление, добавление, изменение (например, вероятности атак) (рисунок 20). Зеленые стрелки на рисунке 20 обозначают добавление, красные – удаление, синие – изменение. Жирные стрелки используются для связи статических контрмер и влияния на граф атак. Пунктирные и сплошные стрелки показаны на рисунке 20 для выделения путей, соответствующих определенным контрмерам, например, открытие порта обуславливает добавление дуги, но не узла.

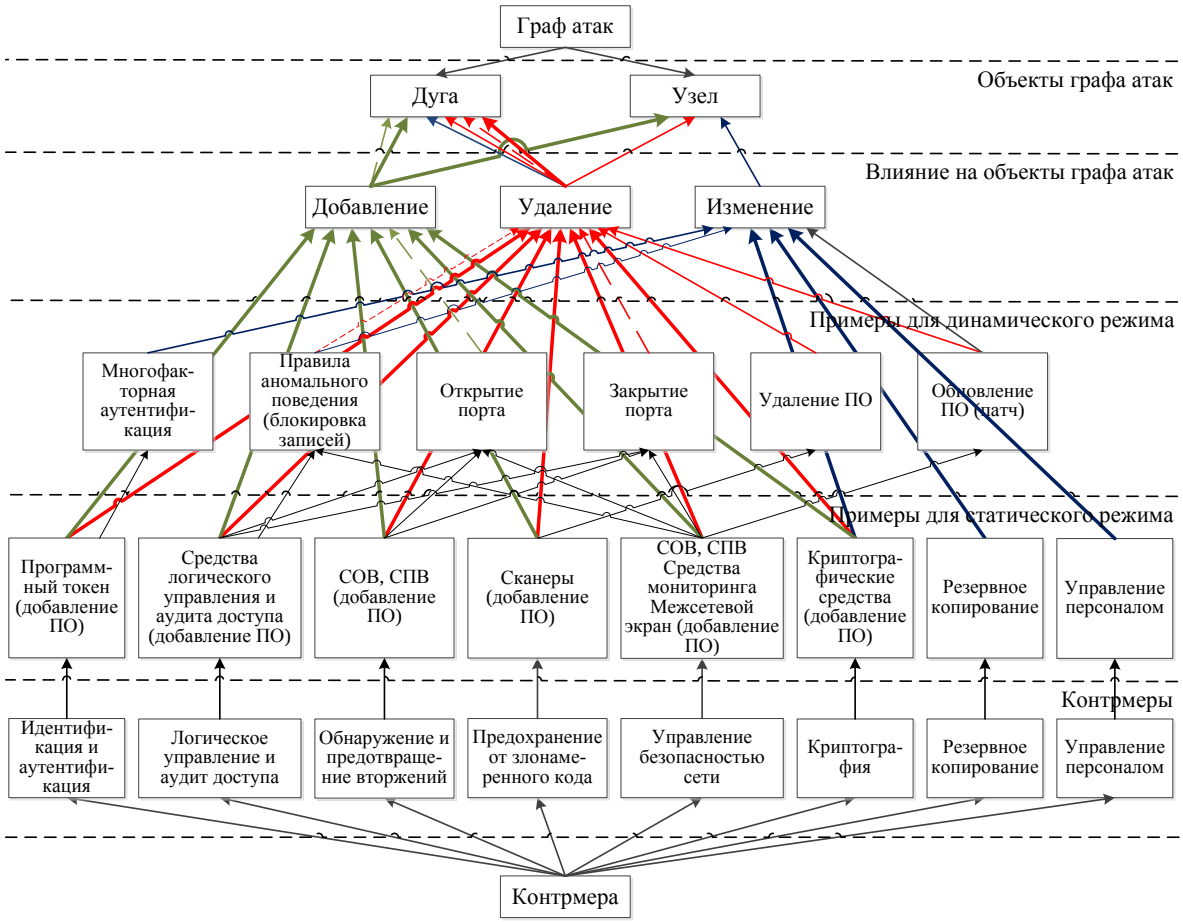


Рисунок 20 – Зависимости между контрмерами и объектами графа атак

Модель контрмер концептуально задается следующим образом: $C=(V, P, M, Sc, AI, SI)$, где V – уязвимость, против которой направлена защитная мера, P – платформа или конфигурация в которой применима защитная мера, M – режим работы системы (статический или динамический), Sc – область действия (элемент графа атак/хост/подсеть/сеть), AI – влияние на граф атак, SI – влияние на граф зависимостей сервисов (удаление, добавление, изменение).

Согласно разделу 2.1, основные показатели уровня выбора контрмер: *уровень побочного ущерба*; *эффективность контрмеры*; *стоимость контрмеры*.

Показатель *уровень побочного ущерба* определяется в виде трехмерного вектора $\left[CD_c \quad CD_i \quad CD_a\right]$, где CD_c, CD_i, CD_a – ущерб для свойств конфиденциальности, целостности и доступности, соответственно, в результате реализации контрмеры, принимают значения от 0 до 1. Показатель определяется на основе графа зависимостей сервисов: вначале определяется сервис (группа сервисов), который затрагивается контрмерой (на основе информации о скомпрометированном хосте из события безопасности); затем определяется критичность данного сервиса (полученная на основе графа зависимостей сервисов); полученная критичность умножается на уровень достоверности события безопасности.

Показатель *эффективность контрмеры* определяет степень исправления свойства безопасности в виде трехмерного вектора $\left[CE_c \quad CE_i \quad CE_a\right]$, где CE_c, CE_i, CE_a – значения эффективности исправления свойств конфиденциальности, целостности и доступности, соответственно, в результате реализации контрмеры, принимают значения от 0 до 1. Задается вручную администратором.

Показатель *стоимость контрмеры* определяет стоимость реализации контрмеры, измеряется в денежных единицах. Задается вручную администратором.

Алгоритмы вычисления интегральных показателей. К интегральным показателям относятся *уровень риска атаки*, *уровень риска сервиса*, *уровень риска хоста*, *уровень риска КС*, *индекс выбора контрмеры*.

Как было сказано выше, в данном исследовании под риском понимается комбинация факторов вероятности возникновения инцидента (или проведения атаки) и его разрушительного воздействия [6]. Входные данные алгоритма определения уровня риска: модели входных данных соответствующего уровня, в том числе модель КС и модель атак; показатели защищенности соответствующего уровня, в том числе, *ущерб от атаки* и *вероятность атаки*. Алгоритм включает этапы: сбор входных данных; определение уровня вычислений; вычисление значения риска. Выходные данные алгоритма: *риск* соответствующего уровня.

Алгоритм вычисления уровня риска на топологическом уровне. Для определения уровня риска на топологическом уровне предлагается использовать оценки CVSS для уязвимостей. Уравнения CVSS позволяют учитывать ущерб, наносимый уязвимостью, и вероятность ее использования, что удовлетворяет определению риска, данному в [10]. Согласно методике оценки защищенности КС, предлагаемой в данной работе, на

топологическом уровне не учитывается последовательное применение нескольких уязвимостей для реализации многошаговых атак, поэтому отсутствие корреляции между уязвимостями в уравнениях CVSS не является ограничением. Уровень риска предлагается определять на основе модифицированного контекстного уравнения CVSS, так как оно позволяет учитывать связь между оценкой уязвимости и критичностью активов. Контекстное уравнение [109]:

$$EnvironmentalScore = round\_to\_1\_decimal((AdjustedTemporal + (10 - AdjustedTemporal) \times CollateralDamagePotential) \times TargetDistribution),$$

где *AdjustedTemporal=TemporalScore*, в котором *BaseScore Impact* (ущерб от эксплуатации уязвимости) заменен на *AdjustedImpact*;

TemporalScore – временная оценка по CVSS;

$$AdjustedImpact = \min(10, 10, 41 \times (1 - (1 - ConfImpact \times ConfReq) \times (1 - IntegImpact \times IntegReq) \times (1 - AvailImpact \times AvailReq))),$$

где *ConfImpact, IntegImpact, AvailImpact* – влияние на конфиденциальность, целостность, и доступность, соответственно, в результате эксплуатации уязвимости;

ConfReq, IntegReq, AvailReq – требования безопасности;

CollateralDamagePotential – потенциал побочного ущерба при эксплуатации уязвимости;

TargetDistribution – плотность целей.

Для учета критичности активов в уравнении риска воспользуемся требованиями безопасности *ConfReq, IntegReq* и *AvailReq* (будем рассматривать их как критичность актива). Показатели *TargetDistribution* и *CollateralDamagePotential* учитывать не будем.

Тогда уравнение принимает вид: *Risk*=round_to_1_decimal(*AdjustedTemporal*).

Временная оценка определяется на основе уравнения [109]:

$$TemporalScore = \text{round\_to\_1\_decimal}(BaseScore \times Exploitability \times RemediationLevel \times ReportConfidence),$$

где *Exploitability* – возможность использования уязвимости;

RemediationLevel – уровень исправления уязвимости;

ReportConfidence – степень достоверности отчета об уязвимости.

Чтобы получить показатель *AdjustedTemporal*, заменим *BaseScore Impact* на *AdjustedImpact*. *BaseScore Impact* применяется при расчете *BaseScore*, при замене *BaseScore Impact* на *AdjustedImpact*, получим показатель *AdjustedBase*:

$$AdjustedBase = \text{round\_to\_1\_dicimal}(((0,6 \times AdjustedImpact) + (0,4 \times Exploitability) - -1,5) \times f(AdjustedImpact)),$$

где $f(AdjustedImpact) = \begin{cases} 0, \text{ если } AdjustedImpact = 0 \\ 1,176, \text{ если } AdjustedImpact \neq 0 \end{cases}$;

Exploitability – возможность использования уязвимости.

Показатели *Exploitability, RemediationLevel* и *ReportConfidence* учитывать не будем, тогда: *AdjustedTemporal*=round_to_1_decimal(*AdjustedBase*). Уравнение риска принимает вид: *Risk*=round_to_1_decimal(*AdjustedBase*).

При подстановке получаем:

$$Risk = \text{round\_to\_1\_dicimal}(((0,6 \times AdjustedImpact) + (0,4 \times Exploitability) - 1,5) \times \times f(AdjustedImpact)).$$

Заменим показатели *ConfReq*, *IntegReq* и *AvailReq* в уравнении для *AdjustedImpact* на показатели критичности, тогда:

$$AdjustedImpact = \min(10, 10,41 \times (1 - (1 - ConfImpact \times Criticality(c)) \times (1 - -IntegImpact \times Criticality(i)) \times (1 - AvailImpact \times Criticality(a)))),$$

где *Criticality*(*c*), *Criticality*(*i*) и *Criticality*(*a*) – критичность конфиденциальности, целостности и доступности актива, соответственно.

Возможные значения показателей критичности, и их преобразование для применения в уравнении оценки риска, приведены в таблице 8. Таким образом, риск может принимать значения от 0 до 10.

Таблица 8 – Преобразование оценок критичности актива для применения в уравнении оценки риска

Критичность	[0:0,01)	[0,01:0,1)	[0,1:1)	[1:10)	[10:100)	100
Оценка	0	0,5	1	1,2	1,4	1,51

После того, как определен риск каждой уязвимости хоста, оценка риска для экземпляра программно-аппаратного обеспечения определяется как максимальная из данных оценок, а оценка риска для хоста – как максимальная из оценок для программно-аппаратного обеспечения. Уровень риска для КС в целом определяется максимальной оценкой риска хостов. Уровень риска при этом определяется как «Высокий» / «Средний» / «Низкий» в соответствии с уровнями CVSS оценок. Так можно выделить наиболее незащищенные участки системы.

Разработка данного алгоритма включала выделение показателей, применяемых для вычисления уровня риска, преобразование уравнения CVSS для включения показателя критичности, преобразование шкалы значений показателя критичности для включения в уравнение CVSS, формирование правил определения уровня риска для различных объектов КС (ПО, хостов, КС) на основе уровня риска уязвимостей.

Алгоритм вычисления уровня риска на уровне графа атак, атакующего и событий. Для этих уровней *риск* определяется по формуле: *Risk=AttackImpact*× ×*AttackPotentiality*, где *AttackImpact* – ущерб от атаки (комбинация разрушительности атакующего действия и критичности актива); *AttackPotentiality* – вероятность атаки. *AttackImpact* и *AttackPotentiality* определяются на основе алгоритмов соответствующего уровня.

Риск определяется для узлов графа атак (путем произведения показателей вероятности и ущерба соответствующего узла). Значение риска варьируется от 0 до 100, так как минимальное значение вероятности и ущерба 0, а максимальное значение вероятности 1 и ущерба – 100. Риск от 0 до 0,1 принимается низким (то есть риском можно пренебречь), риск от 0,1 до 1 – средним (меры необходимо принять), риск от 1 до 10 – высоким (меры необходимо принять как можно скорее), а от 10 до 100 – критическим (меры необходимо принять немедленно).

Риск для атаки определяется как произведение вероятности для последнего из последовательности узлов атаки на графе (минимальной) на суммарный ущерб по всем узлам атаки. *Риск для хоста* определяется как максимальный из рисков всех атак, проходящих через хост. *Риск для КС* определяется как максимальный из рисков хостов.

Алгоритм вычисления индекса выбора контрмеры. Показатель *индекс выбора контрмеры CI* определяется на основе выигрыша в результате реализации контрмеры и затрат на ее реализацию (а именно, стоимости реализации контрмеры и побочного ущерба от ее реализации) [91]. Для этого определяются возможные потери до ($Risk_a$) и после ($Risk_b$) реализации контрмеры. Выигрыш определяется как разница между $Risk_a$ и $Risk_b$: $Benefit=Risk_a–Risk_b$, где $Risk_a$ являются суммой риска по всем узлам, затрагиваемым контрмерой до ее реализации, а $Risk_b$ – после ее реализации. При этом выигрыш должен стремиться к максимально возможному значению, а затраты – к минимальному. Поэтому *индекс выбора контрмеры CI* определим как:

$$CI = \frac{Benefit}{CD + CC} = \frac{Risk_a - Risk_b}{CD + CC},$$

где *CD – побочный ущерб*,

CC – стоимость контрмеры.

При выборе контрмер необходимо максимизировать данный показатель.

Анализ индекса выбора контрмеры. Рассмотрим граничные значения параметров, применяемых для вычисления данного показателя. В случае если *Benefit*<=0 показатель принимает значение меньшее или равное 0, поэтому на значение данного показателя накладывается ограничение: *CI*>0, так как отсутствие выигрыша не имеет смысла, а отрицательное значение показателя говорит о том, что реализация контрмеры приведет к еще большим потерям, чем атака. Не учитывается ситуация, когда стоимость реализации контрмеры (*CD*+*CC*) равна 0. В дальнейшем эту проблему можно будет решить например на основе идеи, реализованной в [74], путем учета стоимости инфраструктуры. Кроме того, необходимо учитывать, что в случае если выигрыш и затраты намного меньше единицы (стремятся к минимальным значениям), и выигрыш и затраты намного больше единицы (стремятся к максимальным значениям), индекс может принимать одинаковые значения. В этом случае можно задавать дополнительное условие (что важнее – максимально снизить риск, или минимизировать затраты).

2.4 Методика выбора защитных мер

Методика выбора контрмер включает модель контрмеры, связь модели контрмер и графа атак, алгоритм вычисления *индекса выбора контрмеры*.

В данном исследовании выделяются статический и динамический режимы работы методики выбора контрмер. Методика статического режима применяется на этапах проектирования и развертывания системы, методика динамического режима – на этапе эксплуатации. В статическом режиме выбираются защитные меры, которые позволят повысить общий уровень защищенности системы. В динамическом – меры, которые позволят остановить атаку. Примеры контрмер: патч для уязвимости (информацию можно взять, например, из базы xForce [146], которая содержит временные оценки CVSS, в том числе *уровень исправления*, который определяет наличие патча для уязвимости); удаление уязвимого ПО; закрытие порта; добавление дополнительных защитных средств (например, межсетевой экран или антивирус).

Списки контрмер, оценки их эффективности и стоимость определяются экспертами.

Методика выбора контрмер в статическом режиме на топологическом уровне. Входные данные для методики выбора контрмер на топологическом уровне: набор доступных контрмер; модель анализируемой КС (включая программно-аппаратное обеспечение и его уязвимости, дающие высокий уровень риска); показатели топологического уровня (включая риск уязвимостей, ПО и хостов; ущерб, наносимый эксплуатацией уязвимостей, по свойствам конфиденциальности, целостности и доступности).

Вначале контрмеры делятся по области воздействия (хосты, программно-аппаратное обеспечение и уязвимости) и свойству воздействия (конфиденциальность, целостность, доступность, и все). Выбор контрмер осуществляется для каждого хоста сети с использованием *индекса выбора контрмеры*. Для выбора оптимального набора контрмер осуществляется перебор с учетом области и свойств воздействия контрмер, исходя из предположения, что максимизация индекса по подобласти воздействия ведет к максимизации индекса по области в целом.

Методика выбора контрмер на топологическом уровне реализуется следующим образом (схема алгоритма работы методики представлена в приложении Д). Для активов (хостов) с неприемлемым уровнем риска, то есть «Высокой» контекстной CVSS оценкой (от 7 до 10), выполняются шаги (предполагается, что для защищенности системы необходимо принять меры против уязвимостей, создающих высокий уровень риска, но эти требования могут быть как повышены, так и понижены):

1) Выбираются контрмеры, имеющие наибольшую область воздействия (в данном случае это хостовые контрмеры, например, «удаление хоста») с учетом свойств воздействия (все; два свойства; одно свойство; ни одного свойства). Свойство воздействия в модели контрмер заложено в оценках показателей эффективности [CE_c CE_i CE_a]: если $CE_c \neq 0$, то контрмеру можно использовать против нарушения конфиденциальности; $CE_i \neq 0$ – против нарушения целостности; $CE_a \neq 0$ – против нарушения доступности.

2) Для выбранных контрмер считается *индекс выбора контрмер* (для этого пересчитывается *риск* для хоста в случае применения контрмеры, *стоимость ее реализации* и *побочный ущерб*). Обозначим индекс в случае воздействия на все свойства как *И1*, индекс для остальных случаев – *И2*.

3) Если есть хостовые контрмеры, влияющие не на все свойства безопасности, рассматривается применение контрмер, область действия которых распространяется на отдельные экземпляры программно-аппаратного обеспечения (например, «удаление ПО» или «обновление ПО»). Для всех экземпляров программно-аппаратного обеспечения (кроме ПО, уязвимости которого наносят ущерб только тем свойствам безопасности, против которых уже выбраны контрмеры более высокого уровня на шаге 2) рассматривается вначале применение контрмер, затрагивающих конкретный экземпляр ПО. При этом вычисляется *индекс выбора контрмеры* для выбранной контрмеры $ИЗ_i$ (i – экземпляр ПО). Данный индекс запоминается, последующие шаги выполняются для всех уязвимостей высокого риска данного экземпляра ПО (кроме уязвимостей, наносящих ущерб только тем свойствам безопасности, против которых уже выбраны контрмеры более высокого уровня):

а) Для всех уязвимостей экземпляра ПО, формирующих высокую оценку риска, определяется показатель CVSS у*ровень исправления* (*RemediationLevel*). В случае наличия исправления, считается *индекс выбора контрмеры* для контрмеры «исправление уязвимости». При этом риск пересчитывается на основе временного уравнения CVSS *TemporalScore*=round_to_1_decimal(*BaseScore*×*RemediationLevel*), где функция round_to_1_decimal выполняет округление аргумента до одного знака после запятой. Суммарный показатель *индекса выбора контрмеры* по контрмере «исправление уязвимости» вычисляется на основе максимального *риска* по уязвимостям и суммарной *стоимости*.

б) Если все уязвимости экземпляра ПО рассмотрены, для экземпляра ПО выбирается максимальный *индекс выбора контрмеры* (и соответствующие контрмеры) из контрмер по уровню ПО и уровню уязвимостей и принимается за *ИЗ*.

4) Вычисляется общий *индекс выбора контрмеры* по всем экземплярам ПО, с учетом контрмер, выбранных для каждого экземпляра, выбором максимального риска среди всех *ИЗ* и суммарной стоимости, и принимается за *ИЗ*.

5) Вычисляется общий *индекс выбора контрмеры* по свойству выбором максимального риска из *И2* и *ИЗ* и суммарной стоимости. Обозначим его *И4*.

6) Для хоста выбирается максимальный индекс из *И1* и *И4*. Соответствующие контрмеры выбираются в качестве выходного набора для данного хоста.

Выходные данные работы методики: набор оптимальных контрмер для всех хостов.

Методика выбора контрмер в статическом режиме на уровне графа атак и атакующего. Входные данные для методики выбора контрмер на уровне графа атак: набор доступных контрмер; модель анализируемой КС; граф атак; показатели уровня графа атак (включая риск узлов графа; ущерб, наносимый эксплуатацией уязвимостей).

Методика выбора контрмер на уровне графа атак реализуется в несколько этапов (алгоритм работы методики представлен на рисунке 21):

1) Включить в набор точек для применения контрмер все входные узлы графа (изначально доступные атакующему).

2) Определить узлы графа, представляющие наибольший риск. Для этого применяется алгоритм, предложенный в [57], адаптированный для графа атак, применяемого в данном исследовании (исходный алгоритм работает для отдельных уязвимостей, в данном исследовании он применяется для групп уязвимостей, соответствующих узлам графа атак). Схема адаптированного алгоритма приведена в приложении Е.

3) Включить в набор точек для применения контрмер узлы графа, через которые проходит наибольший суммарный риск.

4) Применить алгоритм выбора контрмер топологического уровня для выбранных узлов с учетом того, что: область действия контрмер определяется по объектам графа атак (подграф, дуга, узел, уязвимость); для переопределения уровня риска необходимо перестроить граф атак.

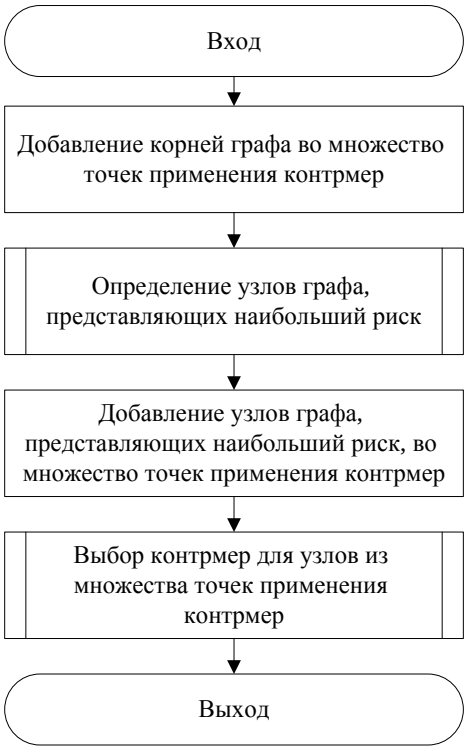


Рисунок 21 – Алгоритм выбора контрмер на уровне графа атак

На *уровне атакующего* дополнительно учитываются возможности атакующего. Отличие от уровня графа атак состоит только в изменении значений *риска* для узлов графа и изменении множества корней графа.

Методика выбора контрмер в динамическом режиме. Методика *уровня событий* (рисунок 22) отличается тем, что основной целью является предотвращение обнаруженной атаки, а не повышение общего уровня защищенности КС. На уровне событий контрмеры реализуются в зависимости от текущих и будущих (спрогнозированных) шагов атакующего. При этом учитывается «глубина графа до критичного ресурса», которая определяется как количество узлов графа до актива с высоким уровнем критичности. Если данная глубина превышает определенное значение, то система ждет нового события для уточнения своих оценок, если глубина меньше определенного значения, система предлагает контрмеру на основе имеющихся данных со степенью точности, соответствующей количеству уже выявленных релевантных событий.

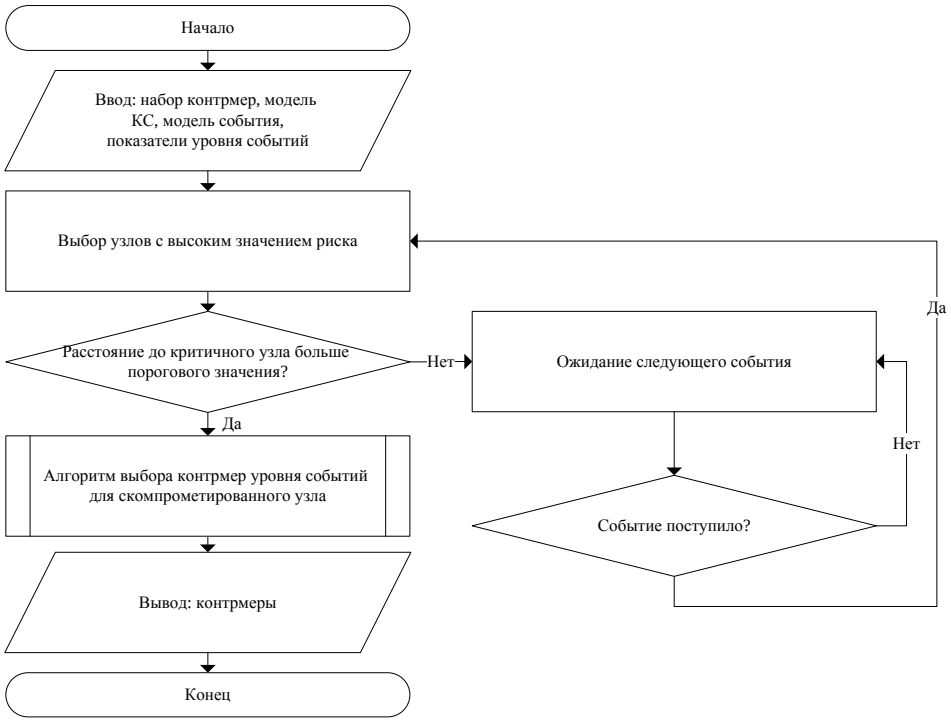


Рисунок 22 – Методика выбора контрмер на уровне событий

Входные данные для методики выбора контрмер на уровне событий: набор доступных контрмер; модель анализируемой КС (включая программно-аппаратное обеспечение и его уязвимости); модели поступивших событий безопасности; показатели защищенности уровня событий.

Методика включает следующие основные этапы (рисунок 22):

1) Выделение узлов, для которых значение риска больше или равно «Высокий» (порог можно изменить, в зависимости от требований администратора).

2) Решение о применении контрмер для выделенных узлов (или об ожидании новых событий) в зависимости от «глубины графа до критичного ресурса».

3) Если принято решение о применении мер, сортировка контрмер по количеству узлов, на которые они повлияют (в случае контрмер, влияющих на равное количество узлов, создается несколько списков контрмер).

4) Определение контрмер для каждого узла на основе полученных списков: сначала выбираются контрмеры, действующие на наибольшее количество узлов графа. Оставшиеся контрмеры выбираются с учетом максимального несовпадения покрытия (то есть узлов, на которые они воздействуют), и так пока не будут охвачены все узлы.

5) Вычисление *индекса выбора контрмер* для полученных на шаге 4 списков. Для этого пересчитываются риски для каждого узла (если остаются риски больше или равно «Высокий», список контрмер отбрасывается), затем выбирается список с максимальным суммарным *индексом выбора контрмер*.

6) Возврат алгоритма на шаг 4 если на предыдущем шаге были отброшены все списки. Формирование новых списков начиная со второй контрмеры в списках шага 4, так пока не будет найдено удовлетворительное решение.

Выводы по главе 2

1) Разработан комплекс показателей защищенности. В отличие от существующих комплексов, уровни классификации выделены в зависимости от обрабатываемых входных данных (таких как знания о сети, ее уязвимостях, атакующем, происходящих в системе событиях безопасности), что позволяет адекватно отражать текущую ситуацию на основе доступных входных данных. Для учета контрмер при оценке защищенности, а также выбора контрмер, добавлен уровень выбора контрмер.

2) Определены показатели каждого уровня комплекса показателей. Показатели каждого уровня делятся на подкатегории: базовые, стоимостные, 0 дня. Это позволяет учитывать при оценке рисков и выборе контрмер стоимостные характеристики и

характеристики атак нулевого дня. Введение стоимостных характеристик позволяет обосновать затраты на реагирование в терминах выигрыша и потерь. Учет характеристик атак нулевого дня позволяет проводить более строгую оценку уровеня защищенности сети. Внутри каждой подкатегории выделяются основные показатели, используемые для вычисления значения уровня риска, и вспомогательные показатели, не используемые для вычисления значения уровня риска (предоставляющие дополнительную информацию для оператора).

3) Разработана многоуровневая методика оценки защищенности КС, основанная на предложенном комплексе показателей защищенности. Методика позволяет применять для оценки рисков и выбора контрмер наиболее подходящие с точки зрения доступных входных данных алгоритмы и таким образом получать наиболее адекватную и полную оценку.

4) Разработаны и усовершенствованы модели (в том числе модель атак в виде Байесовского графа атак, модель зависимостей между сервисами сети, модель сервиса, модель атакующего и модель события) и алгоритмы вычисления показателей защищенности. Модели и алгоритмы соответствуют различным уровням предложенной в исследовании методики и учитывают входные данные соответствующего уровня. Модель атак в виде Байесовского графа атак позволяет определить вероятность различных путей атаки и влияние на нее событий безопасности. Модель зависимостей между сервисами сети применяется для автоматизации определения критичности ресурсов сети и связи между бизнес-активами и ИТ активами, что позволяет определять возможные потери в случае компьютерных атак и побочный ущерб при реализации контрмер. Модель контрмеры основана на стандартах, входящих в состав SCAP (ERI и CRE), что позволяет включить их в общий подход оценки и выбора контрмер и автоматизировать заполнение. Модель контрмер применяется в методике выбора контрмер и позволяет оценить эффективность мер защиты с точки зрения противодействия конкретной атаке. Модель событий построена таким образом, чтобы иметь возможность отображения на граф атак и интегрироваться с SIEM-системами.

5) Разработана методика выбора контрмер, основанная на предложенном многоуровневом подходе и показателях защищенности, позволяющая выбрать наиболее адекватные контрмеры с применением максимального количества доступных данных.

Глава 3 Реализация системы оценки защищенности и выбора контрмер и оценка ее эффективности

3.1 Архитектура и реализация программного прототипа системы оценки защищенности компьютерных сетей и выбора защитных мер

Разработанные методики и алгоритмы реализованы в рамках системы оценки защищенности КС и выбора контрмер (СОЗВК), которая является частью системы анализа защищенности на основе графов атак, реализованной в рамках SIEM-системы, разработанной в проекте MASSIF [105]. Программный прототип системы позволяет оценить эффективность предложенных методик оценки защищенности и выбора контрмер. Согласно постановке задачи СОЗВК должна реализовывать функции: (1) получение адекватной и актуальной оценки защищенности КС на основе доступных входных данных в статическом и динамическом режимах работы; (2) учет характеристик атакующего; (3) учет взаимосвязей между сервисами КС для тщательного учета возможного распространения ущерба в случае успешной реализации атак, или побочного ущерба при реализации защитных мер, в статическом и динамическом режимах работы; (4) учет стоимостных характеристик атак и защитных мер, для того, чтобы определить выигрыш в случае реализации защитных мер, в статическом и динамическом режимах работы; (5) автоматизация процесса представления и обработки данных, применяемых для оценки защищенности и выбора защитных мер, в статическом и динамическом режимах работы; (6) выбор наиболее адекватного решения по реагированию с учетом стоимостных требований в статическом и динамическом режимах работы; (7) выявление слабых мест КС в статическом режиме работы; (8) выявление возможных атак на КС, и получение набора показателей защищенности, характеризующего их, в статическом режиме работы; (9) выбор средств защиты для повышения уровня защищенности системы, в статическом режиме работы; (10) учет событий безопасности, происходящих в КС, и переоценка ситуации по защищенности в соответствии с полученной информацией, в динамическом режиме работы; (11) интеграция с SIEM-системами в динамическом режиме работы; (12) выбор защитных мер для предотвращения развивающейся атаки, в динамическом режиме работы.

Архитектура системы представлена на рисунке 23. Компонент обработки данных получает входные данные от администратора, компонента сбора информации (который получает входные данные от сенсоров, сетевых сканеров, хостовых программных агентов, SIEM-системы, и обрабатывает получаемые данные), компонента моделирования атак, и генерирует модели входных данных. Полученные модели применяются как входные данные для компонента оценки защищенности. Компонент оценки защищенности включает набор функций, реализующих методики вычисления показателей защищенности уровня, соответствующего получаемым моделям, и методику оценки защищенности. В случае поступления новых данных показатели пересчитываются. Далее полученные показатели применяются для выбора контрмер.

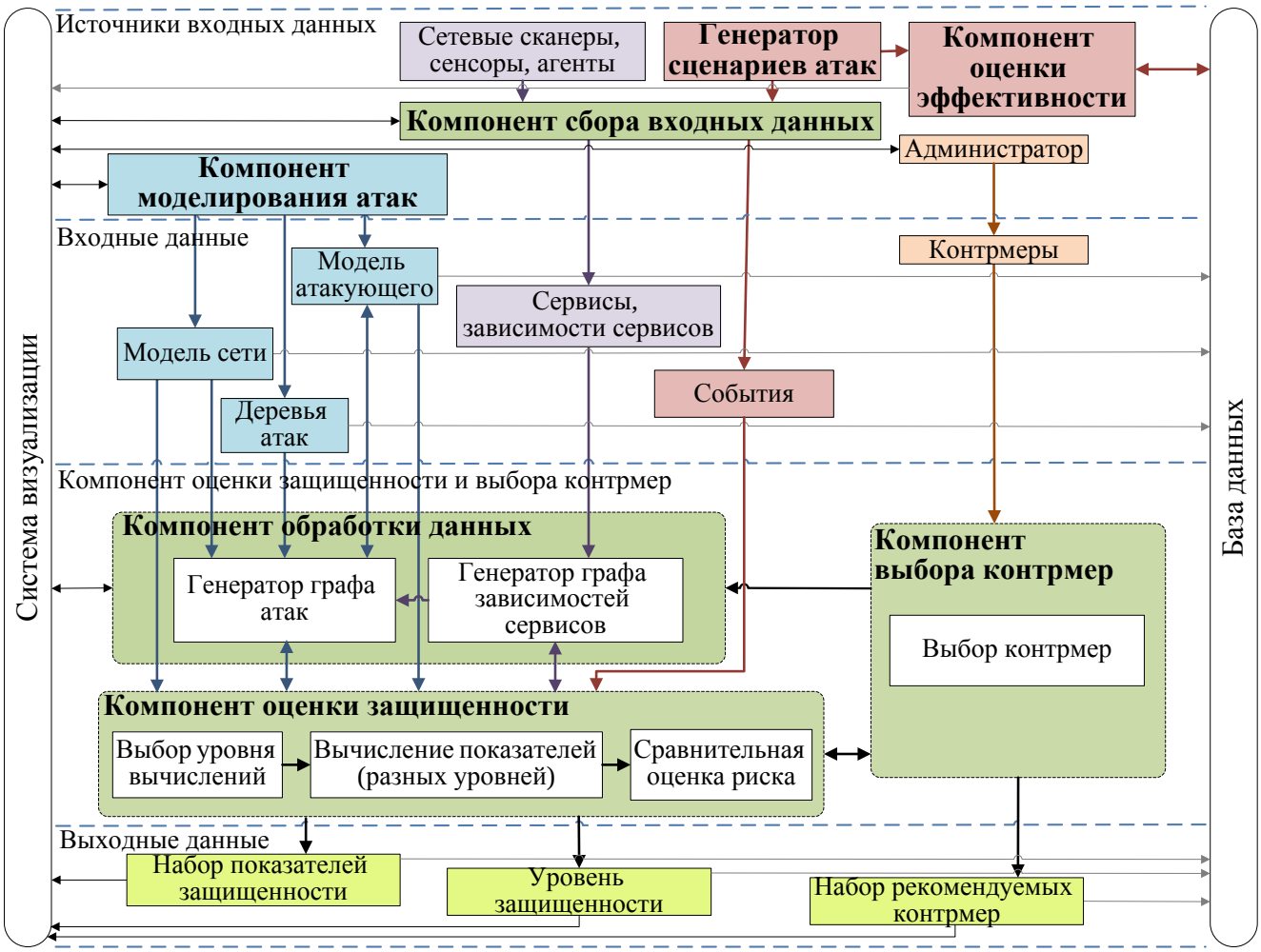


Рисунок 23 – Архитектура системы оценки защищенности и выбора контрмер

Выходные данные системы включают уровень защищенности КС, вычисленные показатели защищенности и набор контрмер. Выходные данные передаются для

отображения системе визуализации. Ниже подробнее рассмотрены основные компоненты СОЗВК.

Система визуализации, *компонент моделирования атак* и *база данных* являются внешними системами по отношению к СОЗВК. *Система визуализации* позволяет пользователю управлять системой, задавать входные данные и просматривать результаты работы системы [29]. *Компонент моделирования атак* [43] формирует исходное дерево атак на основе которого генерируется Байесовский граф атак, применяемый в СОЗВК для оценки защищенности. В *базе данных* [43] хранятся конфигурации анализируемой КС, реализуемая в сети политика безопасности (общие правила функционирования сети), события, происходившие в системе (зафиксированные действия атакующего), возможные и реализованные контрмеры и результаты их внедрения. *Компонент обработки данных* объединяет функции формирования моделей на основе получаемых входных данных: (1) генератор графа зависимостей сервисов преобразует зависимости между сервисами в граф; (2) генератор графа атак преобразует дерево атак в Байесовский граф атак. *Компонент оценки защищенности* содержит функции расчета предложенных в исследовании показателей защищенности. Он включает функцию анализа имеющихся входных данных, необходимую для определения уровня вычислений и функцию сравнительной оценки риска, необходимую для определения уровня защищенности КС. *Компонент выбора контрмер* содержит функции выбора контрмер статического и динамического режимов, и компонент анализа входных данных (для определения уровня вычислений). *Генератор сценариев атак* и *компонент оценки эффективности* разработаны для оценки эффективности разработанной СОЗВК. *Генератор сценариев атак* содержит функции генерации последовательностей атак, приближенных к реальным действиям атакующего в системе и функции генерации последовательностей соответствующих им событий. *Компонент оценки эффективности* содержит функции сравнения полученных в СОЗВК результатов с требованиями, поставленными в главе 1.

Для практической оценки предложенного подхода разработан прототип СОЗВК. Функциональная схема прототипа СОЗВК представлена на рисунке 24. Прототип был реализован на языке Java с использованием принципов объектно-ориентированного программирования, на ОС Microsoft Windows, Intel Core i5 CPU и 12 GB ОЗУ.

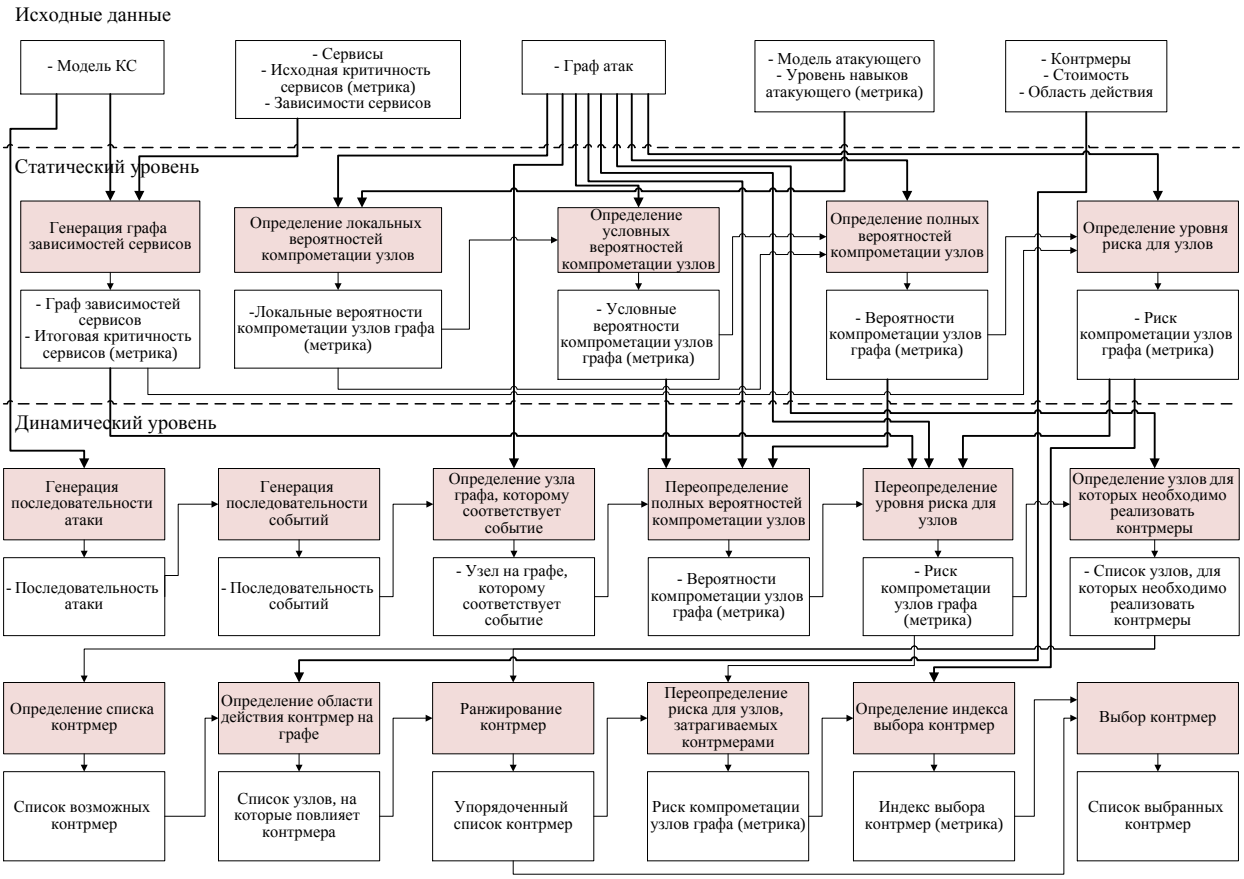


Рисунок 24 – Функциональная схема прототипа системы оценки защищенности и выбора контрмер

Графический интерфейс пользователя программного прототипа СОЗВК представлен на рисунке 25.

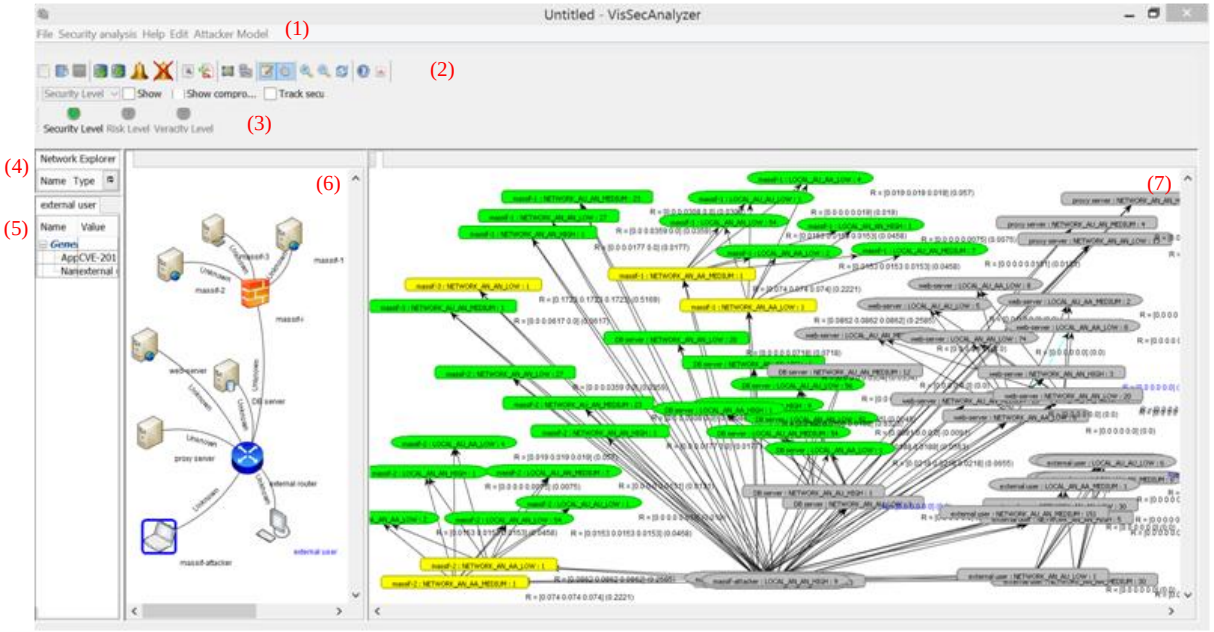


Рисунок 25 – Графический интерфейс пользователя программного прототипа СОЗВК

Графический интерфейс пользователя содержит: (1) главное меню; (2) панель управления; (3) панель графического представления уровня защищенности; (4) диалог доступа к сети (Network Explorer); (5) диалог свойств сети (Property Explorer); (6) окно представления конфигурации сети; (7) окно представления графа атак.

3.2 Генератор сценариев атак на компьютерную сеть

Для оценки эффективности разработанного прототипа был разработан дополнительный инструмент — генератор сценариев атак на КС [93]. Генератор позволяет автоматически генерировать входные данные, близкие к реальным действиям атакующего во время проведения атак на КС в виде последовательностей шаблонов атак на основе шаблонов CAPEC [59]. Также генератор позволяет формировать соответствующие данным сценариям последовательности событий безопасности. В основе генератора лежит подход «черного ящика», при котором предполагается, что у атакующего нет никакой информации об атакуемой сети, и он должен провести все традиционные шаги атаки (разведка, зондирование, проникновение, заметание следов). На рисунке 26 представлена обобщенная архитектура генератора сценариев атак.

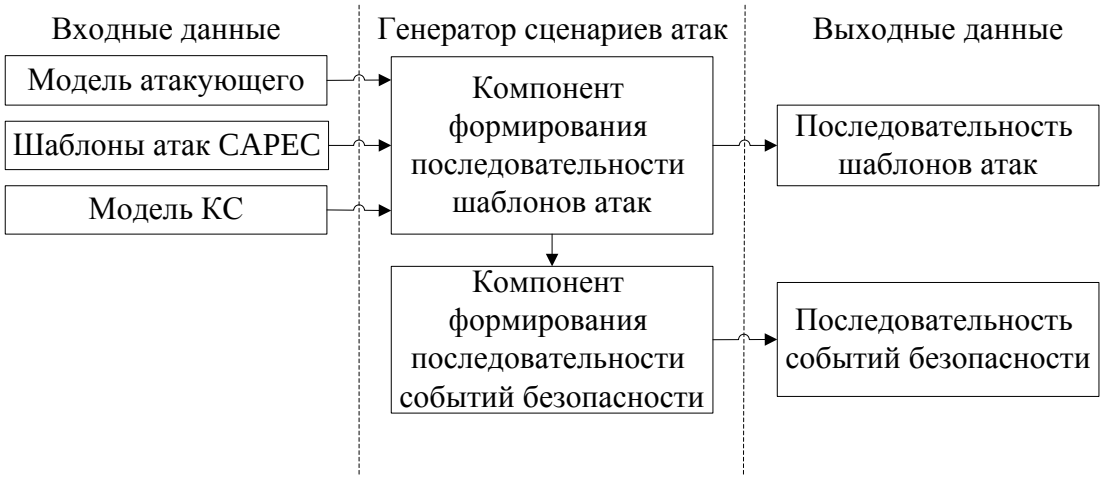


Рисунок 26 – Обобщенная архитектура генератора сценариев атак

В качестве входных данных используется следующая информация: модель КС; информация о ПО хостов сети (в виде идентификаторов CPE [59]) и уязвимостях хостов сети (в виде идентификаторов CVE [61]); модель атакующего, включая информацию о его расположении в сети, знаниях о сети и уровне навыков; шаблоны атак CAPEC [59].

CAPEC поддерживает три способа представления атак: иерархическое представление по механизмам атак и по областям атак (граф), представление по связям с внешними факторами (внешний срез), представление по связям с определенными атрибутами (внутренний срез). Каждый уровень представления содержит ряд категорий атак. Каждая категория содержит структурированные данные о шаблонах атак, включая описание, методы реализации атаки, критичность, примеры, связи с элементами CVE [61] и другие поля. При реализации генератора использовалось представление CAPEC по областям атаки, так как данный вид классифицирует шаблоны по средам, к которым они применимы. Оно включает шесть основных категорий атак: социальная инженерия (social engineering), канал поставок (supply chain), коммуникации (communications), ПО (software), физическая безопасность (physical security), аппаратное обеспечение (hardware). На данный момент генератор учитывает шаблоны только из категории software (что позволяет связать шаблоны с топологией КС).

Алгоритм формирования последовательности шаблонов атак работает следующим образом. На вход алгоритма поступают модель атакующего, модель КС, и шаблоны атак. В дальнейшем при выборе шаблонов атак, входящих в цепочку атакующих действий, учитываются только шаблоны, по сложности соответствующие уровню навыков выбранной модели атакующего, или меньшей сложности, на основе поля CAPEC *Attacker Skill or Knowledge Required* (возможные значения: «Low», «Medium», «High»). Если, в соответствии с моделью, атакующий обладает информацией о топологии сети, то для него не выбираются шаблоны, позволяющие осуществлять разведывательные действия. В противном случае, далее выбираются шаблоны атаки, для которых значение поля *Purpose* – «разведка». Данный шаг выполняется при любом переходе между хостами.

На основе позиции атакующего в сети и топологии сети определяются хосты, доступные для атаки. Далее выбираются шаблоны атаки, позволяющие получить доступ к одному из доступных хостов на основе полей *Consequence Scope* и *Consequence Technical Impact*. Выбор делается на основе следующих значений этих полей: «Confidentiality», «Availability», «Integrity», «Gain privileges/assume identity». Получению доступа соответствует значение: «Gain privileges/assume identity». После получения доступа выбираются шаблоны, позволяющие повысить привилегии (значение «Gain privileges/assume identity»), или нанести максимальный ущерб (значения

«Confidentiality», «Availability», «Integrity»). Процесс выбора шаблона атаки представлен на рисунке 27.

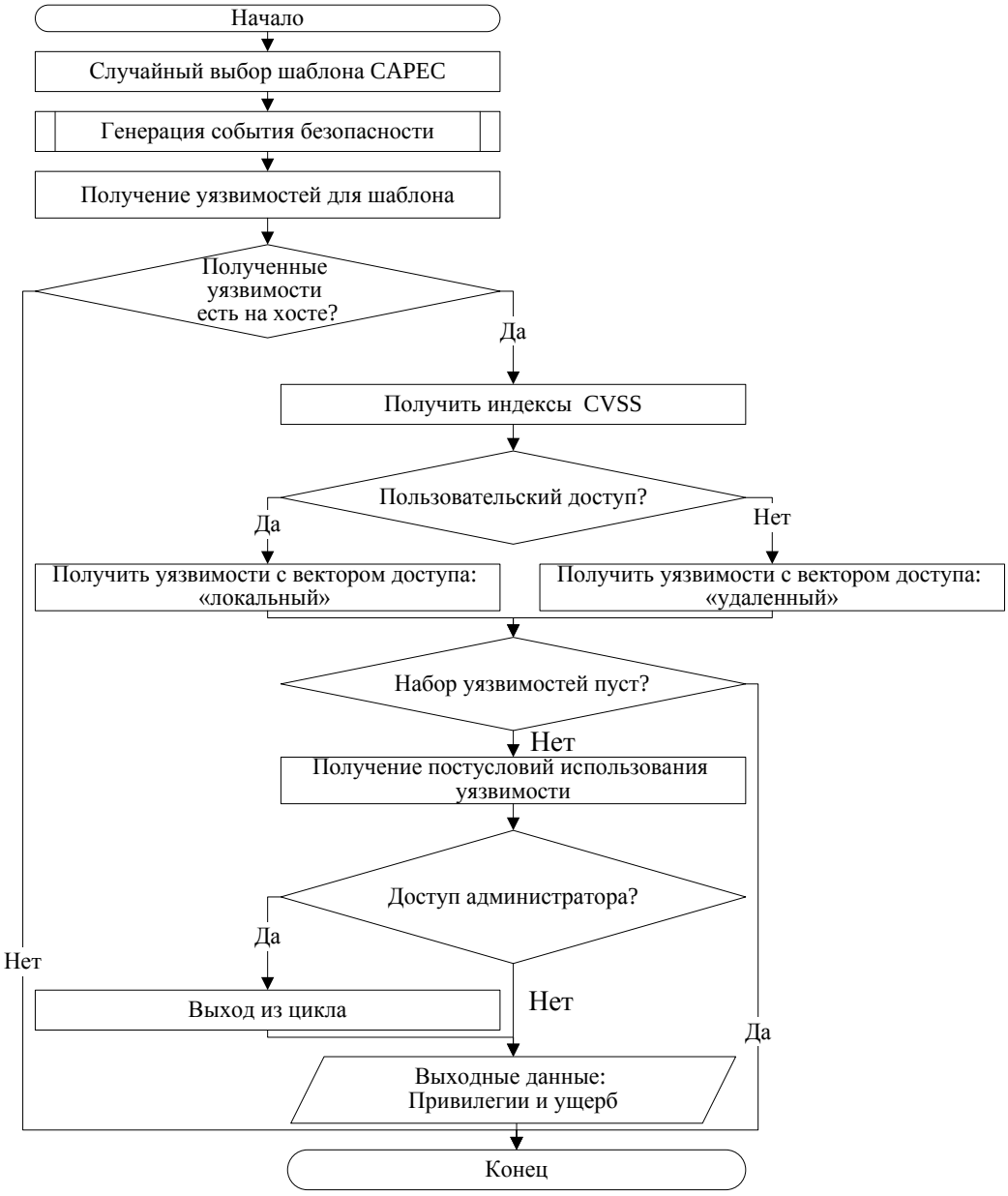


Рисунок 27 – Алгоритм выбора шаблона атаки

Связи между шаблонами атак формируются на основе пред- и постусловий. Которые определяются на основе уязвимостей атакуемого хоста и уязвимостей, соответствующих текущему шаблону атаки (определяются на основе поля *Related Vulnerabilities*). Если одна из уязвимостей, связанных с шаблоном, есть на атакуемом хосте, то атака считается успешной, а последствия успешной эксплуатации уязвимости (ущерб свойствам безопасности и/или полученные привилегии) – постусловиями шаблона атаки. В противном случае атака считается безуспешной и выбирается другой шаблон атаки.

Кроме того, генератор позволяет формировать последовательности событий безопасности, соответствующие сгенерированной последовательности шаблонов атак, на основе полей CAPEC. При формировании событий безопасности (не случайных, а событий, которые могут возникнуть в случае попытки реализации определенной атаки) учитывается успешность атак. Если шаблон был успешно применен, то событие сообщает об успешном проникновении на соответствующий хост, эксплуатации или разведке, в противном случае сообщается о попытке атакующего действия (при анализе защищенности это помогает определить положение атакующего в системе).

Алгоритм формирования последовательности событий безопасности представлен на рисунке 28. Для генерации последовательностей событий безопасности используется поле CAPEC *Indicators-Warnings of Attack*. Все возможные значения данного поля делятся на следующие три группы (в зависимости от цели атаки *Purpose*): проникновение, эксплуатация, разведка. Возможные значения каждой группы приведены в приложении Ж (таблица Ж.1).

Для проведения экспериментов, программный прототип генератора был вписан в СОЗВК. Место компонента в СОЗВК было представлено на рисунке 23.

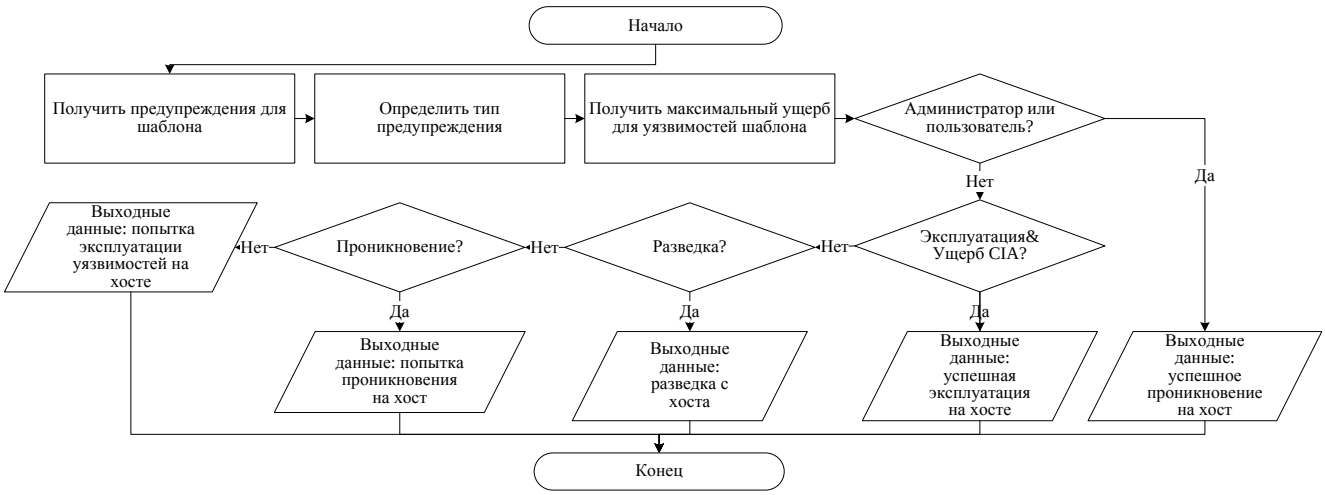


Рисунок 28 – Алгоритм формирования последовательности событий безопасности

3.3 Оценка сложности разработанных алгоритмов и эффективности применения предложенных методик оценки защищенности КС и выбора контрмер

Качество алгоритмов оценивают по трем свойствам: *определенность*, *массовость* и *результативность* [37].

Свойство *определенности* состоит в точности и общепонятности шагов алгоритма. В данной работе это свойство обеспечивается приведением схем разработанных алгоритмов и подробного описания преобразования входной информации в выходную в рамках алгоритмов.

Свойство *массовости* определяется применимостью алгоритма к исходной информации, которая может меняться в определенных пределах. Входной информацией для алгоритма генерации Байесовского графа атак являются: модель КС, деревья атак, значения исходных показателей защищенности для назначения локальных вероятностей графа. Корректность модели КС и деревьев атак обеспечивается компонентом моделирования деревьев атак [43]. Корректность исходных значений локальных вероятностей обеспечивается ограниченностью их значений и конечностью узлов графа. Корректность алгоритма генерации Байесовского графа атак обеспечивается тем, что для его построения применяется широко используемый алгоритм обхода в глубину на основе получаемой модели сети и деревьев.

Входной информацией для алгоритмов вычисления показателей защищенности являются: модель КС; модель атакующего; граф зависимостей сервисов; Байесовский граф атак; модель события; исходные значения для вычисления показателей.

Граф зависимостей сервисов определяется множеством сервисов КС и множеством зависимостей между ними (зависимости представлены в форме матриц зависимостей). Для сервисов определены уровни критичности. Для каждой зависимости определены матрицы зависимости критичности. Корректность зависимостей и значений критичности, а также матриц зависимостей, обеспечивается ограничениями на формат зависимостей при их задании, ограничениями на возможные значения критичности и конечностью количества сервисов, для которых их можно задать. Граф зависимостей сервисов является входной информацией для алгоритма определения показателей на основе данного графа, использующего широко известный алгоритм обхода графа в глубину, что обеспечивает выполнение требования массовости для последнего.

Алгоритмы на основе Байесовского графа атак используют обход в глубину данного графа, что обеспечивает выполнение требования массовости для алгоритмов на его основе. Исходные значения для алгоритмов определения вероятностей основаны на широко используемой системе CVSS [109], что обеспечивает корректность этих данных.

Корректность модели атакующего обеспечивается компонентом моделирования атак [43].

Корректность модели события обеспечивается корректностью исходных данных для ее формирования, обеспечиваемой компонентом корреляции событий SIEM-системы.

Входной информацией для алгоритмов выбора контрмер являются: модели контрмер; модель КС; граф зависимостей сервисов; граф атак; значения показателей защищенности соответствующего уровня. Корректность модели контрмер обеспечивается применением для ее формирования общеизвестных открытых стандартов. Значения показателей защищенности ограничены предыдущими алгоритмами. Корректность алгоритма выбора контрмер обеспечивается лежащим в его основе алгоритмом полного перебора с ограничениями и конечностью входных значений. Алгоритм перебора ограничен областью применения контрмер.

Свойство *результативности* состоит в получении результата за конечное число шагов. Алгоритмы вычисления показателей основаны на обходе графа или рекурсивном обходе графа. Их результативность обеспечивается существованием терминального условия (множества, содержащего все вершины графа, в первом случае, и ограничения на глубину обходимого графа, во втором).

Следовательно, алгоритмы, разработанные в рамках диссертационного исследования, обладают всеми необходимыми свойствами, присущими алгоритмам.

Определим сложность алгоритмов вычисления показателей на основе графа зависимостей сервисов *SG*. При оценке сложности учитывалось следующее ограничение, приводящие к повышению вычислительной сложности алгоритмов: любые два сервиса анализируемой КС связаны как минимум одной зависимостью.

Сложность алгоритмов вычисления показателей на основе графа зависимостей сервисов равна сложности алгоритма обхода графа в глубину. Она оценивается как O(|*S*|+|*LS*|) [2], где |*S*| – число сервисов в анализируемой сети (число вершин графа); |*LS*| – общее число зависимостей в сети (число ребер графа); O – «О» большое, описывающее асимптотическое поведение функции.

Определим сложность алгоритма формирования Байесовского графа атак. Для этого используются следующие обозначения: (1) N – анализируемая КС; (2) H –

множество хостов КС; (3) $|H|$ – число хостов в анализируемой КС; (4) $|V|$ – число уязвимостей в анализируемой КС; (5) $|LG|$ – общее число связей в графе атак.

Для построения Байесовского графа атак на основе дерева атак, алгоритм по очереди выполняет действия: (1) формирование групп уязвимостей (соответствующих атакующим действиям) для каждого хоста сети; (2) формирование связей между группами внутри хоста для всех хостов; (3) переопределение связей между узлами графа на основе графа связей сети; (4) формирование таблиц дискретных условных распределений вероятностей для всех узлов графа.

Для первого действия сложность определяется как

$$F_1(N) \le |H| \cdot |V|. \quad (1)$$

Для второго:

$$F_2(N) \le |H| \cdot 6 \text{ (по количеству групп).} \quad (2)$$

Для третьего действия, если сеть является полным графом (худший случай), сложность определяется как:

$$F_3(N) \le (|H| - 1) \cdot |H| \cdot 6. \quad (3)$$

Четвертое действие включает обход всех вершин графа (в глубину) и формирование для каждой вершины матрицы дискретных условных распределений вероятностей. Четвертое действие имеет сложность:

$$F_4(N) \le |LG| + |H| \cdot 6 \cdot |Pn| \cdot 2^{|Pn|}, \quad (4)$$

где $|Pn|$ – количество предков узла графа, которое ввиду ограничений, принятых в главе 2, о монотонности графа (не добавляются связи с узлами того же уровня сети или узлами предыдущего уровня), зависит от количества уровней сети n и количества хостов на каждом уровне nh_k ($|H| = \sum_{k=1}^{n} nh_k$), а также количества групп уязвимостей каждого хоста:

$$F_4(N) \le |LG| + \sum_{k=1}^{n} \left[15 \cdot nh_k \cdot (15 \cdot nh_k \cdot (k-1) + 1) \cdot 2^{(15 \cdot nh_k \cdot (k-1)+1)} \right]. \quad (5)$$

В худшем случае (при максимальном количестве связей) $|LG| = \sum_{k=1}^{(|H|-1)\cdot 15} k$:

$$F_4(N) \le \sum_{k=1}^{(|H|-1)\cdot 15} k + \sum_{k=1}^{(|H|-1)\cdot 15} k \cdot 2^k . \quad (6)$$

Тогда общая сложность алгоритма построения байесовского графа атак в худшем случае:

$$\begin{aligned} & F(N) \le F_1(N) + F_2(N) + F_3(N) + F_4(N) \Leftrightarrow \\ & \Leftrightarrow F(N) \le |H| \cdot |V| + |H| \cdot 6 + (|H| - 1) \cdot |H| \cdot 6 + \sum_{k=1}^{(|H|-1)\cdot 15} k + \sum_{k=1}^{(|H|-1)\cdot 15} k \cdot 2^k \Leftrightarrow . \\ & \Leftrightarrow F(N) \le |H| \cdot (|V| + |H| \cdot 6) + \sum_{k=1}^{(|H|-1)\cdot 15} k + \sum_{k=1}^{(|H|-1)\cdot 15} k \cdot 2^k \end{aligned} \quad (7)$$

При фиксированном количестве уязвимостей, характерном для хостов анализируемой сети, и атакующих, сложность алгоритма экспоненциально зависит от количества хостов в сети.

Определим сложность алгоритма определения *вероятности атаки* на основе графа атак, который включает обход графа в глубину и обход матрицы дискретных условных распределений вероятностей для каждой вершины. В худшем случае она оценивается как $O(\sum_{k=1}^{(|H|-1)\cdot 15} k + \sum_{k=1}^{(|H|-1)\cdot 15} k \cdot 2^k)$.

Определим сложность алгоритма вычисления *риска компрометации узла* на основе графа атак. Она определяется сложностью обхода графа в глубину: $O(|LG| + |H| \cdot 6)$.

Определим сложность алгоритма выбора контрмер. Алгоритм по очереди выполняет действия: (1) обход подграфа узла графа атак, на который отображено событие безопасности и выбор узлов с высоким уровнем риска для реализации контрмер; (2) обход массива полученных узлов и создание массивов узлов, на которые влияют одни и те же контрмеры; (3) создание списка контрмер, отсортированного по количеству охватываемых узлов (если контрмеры охватывают одинаковое количество узлов, считается индекс выбора контрмер); (4) обход полученного списка контрмер и назначение контрмер узлам по очереди пока все узлы не будут охвачены.

Для первого действия сложность равна сложности алгоритма обхода графа в глубину и оценивается как $F_1(N) \le |LG| + |H| \cdot 6$. Для второго: $F_2(N) \le |H| \cdot 6$. Для третьего действия сложность определяется: $F_3(N) \le |Cm| \cdot \log|Cm|$, где $|Cm|$ –

количество контрмер. Четвертое действие имеет сложность: $F_4(N) \le |Cm|$. Тогда, общая сложность алгоритма выбора контрмер:

$$F(N) \le F_1(N) + F_2(N) + F_3(N) + F_4(N) \Leftrightarrow$$
$$\Leftrightarrow F(N) \le |LG| + 12 \cdot |H| + |Cm| \cdot \log|Cm| + |Cm|.$$

При фиксированном количестве контрмер сложность алгоритма прямо пропорциональна количеству хостов в полносвязной КС.

Рассмотрим эксперименты, показывающие зависимость времени, необходимого для оценки защищенности и выбора контрмер от количества хостов в сети, и обоснованность разработанных методик. Для проведения экспериментов использовались спецификации трех компьютерных сетей (сеть 1 – 10 хостов (приложение И, рисунок И.1), сеть 2 – 20 хостов (приложение И, рисунок И.2), сеть 3 – 40 хостов (приложение И, рисунок И.3)), являющихся фрагментами реальной сети Олимпийских Игр 2008 года в Пекине (рисунок 29), с добавлением дополнительных элементов.

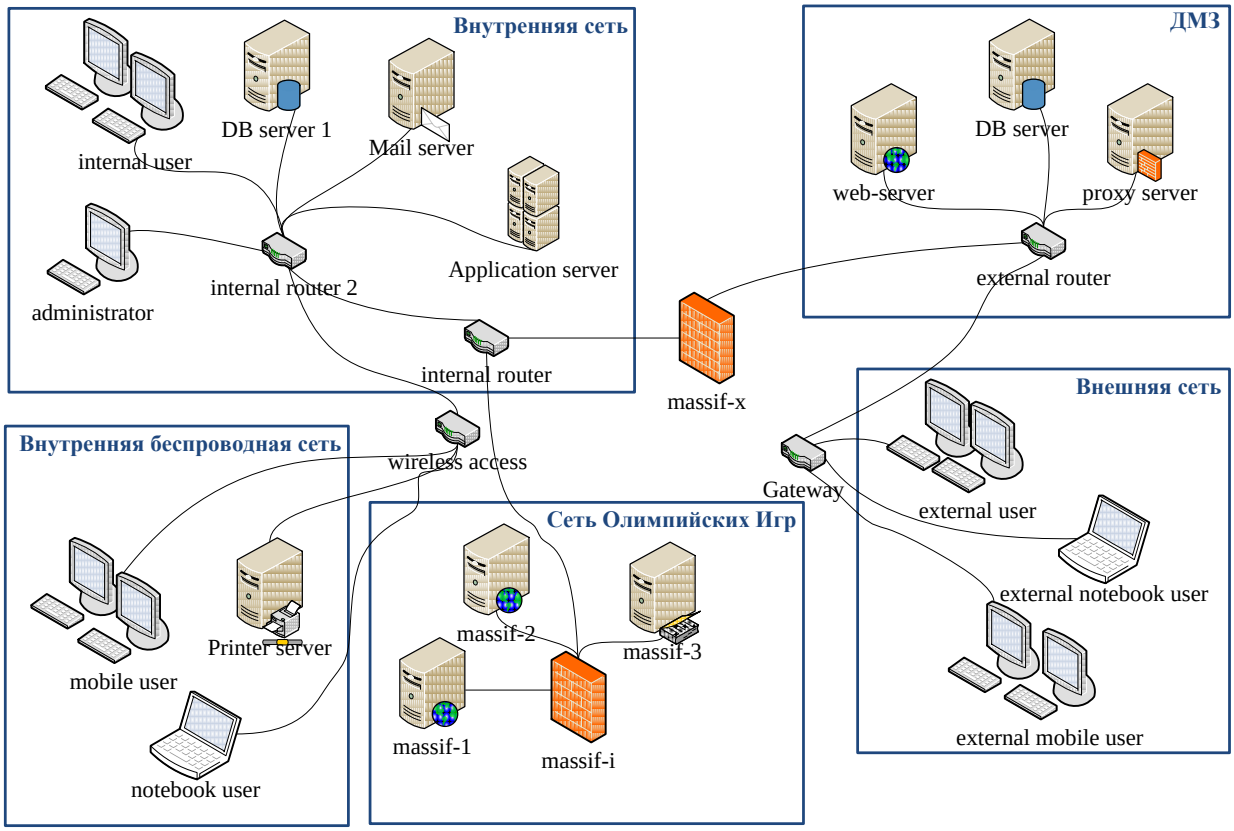


Рисунок 29 – Топология сети Олимпийских Игр 2008 года в Пекине [105]

На рисунке 29 massif-1 и massif-2 – веб-серверы, massif-3 – сервер аутентификации. Эти серверы входят во внутреннюю сеть Олимпийских Игр,

отделенную от внешней сети демилитаризованной зоной (ДМЗ). Топология сети была предоставлена для экспериментов в проекте MASSIF [105] компанией Atos [48].

Для каждой сети средствами сканирования Nmap [117], Nessus [115] и Wireshark [144] определены зависимости между сервисами, которые корректируются администратором (сеть 1 – приложение И, рисунок И.4, сеть 2 – приложение И, рисунок И.5).

Критичности сервисов и веса зависимостей (определенные в рамках проекта MASSIF экспертами) вводятся администратором вручную. Веса зависимостей вводятся в виде трехмерной матрицы, элементы которой соответствуют весу зависимости свойства конфиденциальности, целостности или доступности первого сервиса от свойств конфиденциальности, целостности или доступности второго сервиса. Для сети 1 критичными являются веб-приложения сети Олимпийских Игр. Поэтому им присвоены значения критичности «Серьезная», или 10 по параметрам конфиденциальности, целостности и доступности: [10 10 10]. Веса зависимостей для сети 1 частично приведены в приложении И (таблица И.2). Итоговые критичности сервисов сети рассчитываются на основе методики, описанной в главе 2. Итоговые значения критичностей для сети 1 приведены в приложении И (таблица И.3).

Критичные приложения сети 2 и сети 3 описаны в приложении И. Веса зависимостей сети 2 (отличные от сети 1) частично приведены в приложении И (таблица И.4). Итоговые значения критичностей для сети 2 приведены в приложении И (таблица И.5). Зависимости сервисов сети 3 совпадают с зависимостями сети 2.

На рисунке 30 показано, как отображаются зависимости между сервисами в программном прототипе СОЗВК на примере сети 1. Отображение зависимостей в программном прототипе СОЗВК для сети 2 и сети 3 приведено в приложении К (рисунок К.1).

Для каждой сети проводились эксперименты для трех типов атакующего:

- атакующий 1: внешний (то есть имеет доступ только к хостам, доступным из внешней сети); уровень навыков – High;

- атакующий 2: внешний; уровень навыков – Medium;

- атакующий 3: внешний; уровень навыков – Low.

Цели атакующего будут заданы при формировании экспериментальных цепочек атакующих действий.

Service links: web application (massif-1) - > apacheService (massif-1) AND web application (massif-1) - > authentication service (massif-3) AND web application (massif-1) - > sqlService (DB server) AND web application (massif-1) - > netFilterService (massif-i)

Service links: apacheService (massif-1) - > jbossService (massif-1) AND apacheService (massif-1) - > port http/8080 (massif-1)

Service links: jbossService (massif-1) - > windowsServerService (massif-1) AND jbossService (massif-1) - > port tcp/443 (massif-1)

Service links: web application (massif-2) - > apacheService (massif-2) AND web application (massif-2) - > authentication service (massif-3) AND web application (massif-2) - > sqlService (DB server) AND web application (massif-2) - > netFilterService (massif-i)

Service links: apacheService (massif-2) - > jbossService (massif-2) AND apacheService (massif-2) - > port http/8080 (massif-2)

Service links: jbossService (massif-2) - > windowsServerService (massif-2) AND jbossService (massif-2) - > port tcp/443 (massif-2)

Service links: authentication service (massif-3) - > eDirectoryService (massif-3) AND authentication service (massif-3) - > netFilterService (massif-i)

Service links: eDirectoryService (massif-3) - > slapd service (massif-3)

Service links: slapd service (massif-3) - > port tcp/ldaps 636 (massif-3) AND slapd service (massif-3) - > linuxSuseService (massif-3)

Service links: sqlService (DB server) - > linuxService (DB server) AND sqlService (DB server) - > port tcp/443 (DB server)

Service links: netFilterService (massif-i) - > linuxSuseService (massif-i) AND netFilterService (massif-i) - > appLinuxCitrixService (massif-x)

Service links: appLinuxCitrixService (massif-x) - > linuxKernelService (massif-x)

Рисунок 30 – Отображение зависимостей сервисов для сети 1 в программном прототипе

На рисунке 31 представлен граф атакующих действий для сети 1 для атакующего 1 (графы для сети 2 и сети 3 для атакующего 1 приведены в приложении Л на рисунке Л.1 и рисунке Л.2, соответственно).

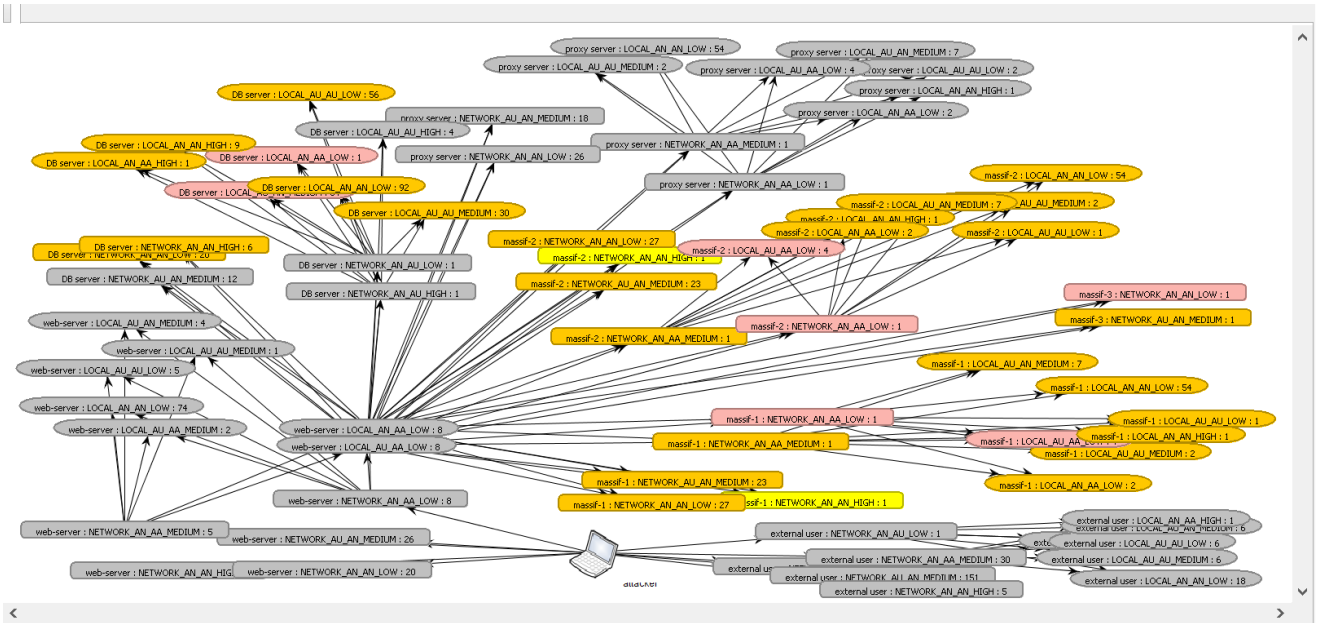


Рисунок 31 – Граф атакующих действий для сети 1 для атакующего 1

Каждый узел графа соответствует группе уязвимостей и задан вектором в формате:

H_NAME: *AccessVector_Authentication_GainedPrivileges_AccessComplexity*,

где H_NAME – название соответствующего хоста,

AccessVector – вектор доступа к уязвимостям группы,

Authentication – требования аутентификации для эксплуатации уязвимостей группы,

GainedPrivileges – привилегии на хосте, получаемые после успешной эксплуатации уязвимостей группы,

AccessComplexity – сложность доступа к уязвимостям группы.

AccessVector, *Authentication*, *GainedPrivileges*, *AccessComplexity* определяются на основе значений в открытой базе уязвимостей NVD аббревиатурами, определенными стандартом CVSS (значения аббревиатур расшифрованы в главе 1).

Узлы выделены красным цветом в случае уровня риска «Критический», оранжевым – в случае уровня риска «Высокий», желтым – в случае уровня риска «Средний», и зеленым – в случае уровня риска «Низкий». Серым цветом выделены узлы с нулевым риском. Значения риска для каждого узла графа приведены в приложении М (таблица М.1).

Для каждого атакующего генерировались цепочки атакующих действий и соответствующих им событий при помощи генератора сценариев атак на КС [93]. На рисунке 32 приведен пример вывода информации о последовательностях атак и событий.

Атаки выводятся в виде последовательности шагов атаки, каждый из которых представлен в формате: *Targeted_application*: *CAPEC-ID*: *Description*, где *Targeted_application* – хост, на который производится атака; *CAPEC-ID* – идентификатор шаблона атаки CAPEC [59]; *Description* – общее описание шага атаки. События выводятся в следующем формате:

Event *Num*: Host *Event_host CAPEC-ID* [*Event_description*],

где *Num* – порядковый номер события; *Event_host* – хост, на котором зафиксировано событие; *CAPEC-ID* – идентификатор шаблона атаки CAPEC [59]; *Event_description* – описание события.

Sequence 19996:
web-server: CAPEC-312: Active OS Fingerprinting
web-server: CAPEC-47: Buffer Overflow via Parameter Expansion
web-server: CAPEC-313: Passive OS Fingerprinting
web-server: CAPEC-14: Client-side Injection-induced Buffer Overflow
massif-2: CAPEC-135: Format String Injection
Events sequence:
Event 1: Host web-server CAPEC-14 [An example of indicator is when the client software crashes after executing code downloaded from a hostile server.]
...
Sequence 19999:
web-server: CAPEC-318: IP 'ID' Echoed Byte-Order Probe
web-server: CAPEC-47: Buffer Overflow via Parameter Expansion
web-server: CAPEC-319: IP (DF) 'Don't Fragment Bit' Echoing Probe
web-server: CAPEC-14: Client-side Injection-induced Buffer Overflow
massif-2: CAPEC-14: Client-side Injection-induced Buffer Overflow
Events sequence:
Event 1: Host web-server CAPEC-14 [An example of indicator is when the client software crashes after executing code downloaded from a hostile server.]
Event 2: Host massif-2 CAPEC-14 [An example of indicator is when the client software crashes after executing code downloaded from a hostile server.]
...
Sequence 20001:
web-server: CAPEC-322: TCP (ISN) Greatest Common Divisor Probe
web-server: CAPEC-10: Buffer Overflow via Environment Variables
web-server: CAPEC-323: TCP (ISN) Counter Rate Probe
web-server: CAPEC-76: Manipulating Input to File System Calls
massif-1: CAPEC-10: Buffer Overflow via Environment Variables
massif-1: CAPEC-324: TCP (ISN) Sequence Predictability Probe
massif-1: CAPEC-10: Buffer Overflow via Environment Variables
Events sequence:
Event 1: Host web-server CAPEC-10 [If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.]
Event 2: Host massif-1 CAPEC-10 [If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.]
Event 3: Host massif-1 CAPEC-10 [If the application does bound checking, it should fail when the data

Рисунок 32 – Пример вывода информации о последовательностях атак и событий

Для каждой сети было сгенерировано более 20 000 последовательностей атак и соответствующих последовательностей событий безопасности. В таблице 9 приведен пример сгенерированных последовательности атак и событий безопасности для сети 1 для атакующего 1. Примеры последовательностей для всех трех сетей приведены в приложении Н (таблица Н.1). Для атакующего 1 при формировании последовательностей атакующих действий не вводилось ограничений на их сложность. Для атакующего 2 выбирались шаблоны CAPEC только среднего и низкого уровня сложности. Для атакующего 3 – только с низким уровнем сложности. Атаки

представлены в виде последовательности шагов атаки, разделенных символом «->». Каждый шаг представлен в формате:

(*Num*) *Description Targeted_application*: *CAPEC-ID*: *CAPEC_Description*,

где *Num* – порядковый номер шага; *Description* – общее описание шага; *Targeted_application* – хост, на который производится атака (если есть); *CAPEC-ID* – идентификатор шаблона атаки CAPEC [59]; *CAPEC_Description* – описание шаблона CAPEC [59]. События представлены в следующем формате:

Событие *Num*: *Event_host* (*CAPEC-ID*): *Event_description*,

где *Num* – порядковый номер события; *Event_host* – хост, на котором зафиксировано событие; *CAPEC-ID* – идентификатор шаблона атаки CAPEC [59]; *Event_description* – описание события.

Для первой сети средняя длина последовательности атаки составила 4 шага, а количество событий безопасности для последовательности атак – 1. Для сети 2 средняя длина последовательности атаки – 6 шагов, а количество событий безопасности – 1. Для сети 3 средняя длина последовательности атаки – 4 шага, а количество событий безопасности – 1. Средняя длина последовательностей атаки почти совпадает, т.к. сети имеют схожую структуру подсетей.

Таблица 9 – Пример последовательности атак и событий безопасности для экспериментов

Последовательность атакующих действий	**Последовательность событий безопасности**
(1) Из внешней сети: CAPEC-170: Web Application Fingerprinting -> (2) Атака на web-server: CAPEC-10: Buffer Overflow via Environment Variables -> (3) Сбор информации с web-server: CAPEC-224: Fingerprinting ->web-server: CAPEC-14: Client-side Injection-induced Buffer Overflow	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert. Событие 2: web-server (CAPEC-14): An example of indicator is when the client software crashes after executing code downloaded from a hostile server.

Рассмотрим влияние событий безопасности на оценку защищенности КС на примере сети 1, атакующего 1 и следующей последовательности атаки: CAPEC-529: Malware-Directed Internal Reconnaissance из внешней сети -> CAPEC-10: Buffer Overflow via Environment Variables на хосте web-server -> CAPEC-541: Application Fingerprinting с хоста web-server -> CAPEC-10: Buffer Overflow via Environment Variables на хосте DB

server -> CAPEC-85: AJAX Fingerprinting с хоста DB server -> CAPEC-10: Buffer Overflow via Environment Variables на хосте DB server.

Сгенерированная последовательность событий для данной последовательности атаки: Событие 1 (хост web-server, шаблон атаки CAPEC-10) «If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.» -> Событие 2 (хост DB server, шаблон атаки CAPEC-10) «If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.» -> Событие 3 (хост DB server, шаблон атаки CAPEC-10) «If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.»

Узлы графа, на которые отображены события безопасности, выделены красной рамкой на рисунке 33. Отметим, что изначально риск компрометации критических узлов сети достаточно высокий. Таким образом, уровень риска позволяет отследить наиболее критичные узлы сети.

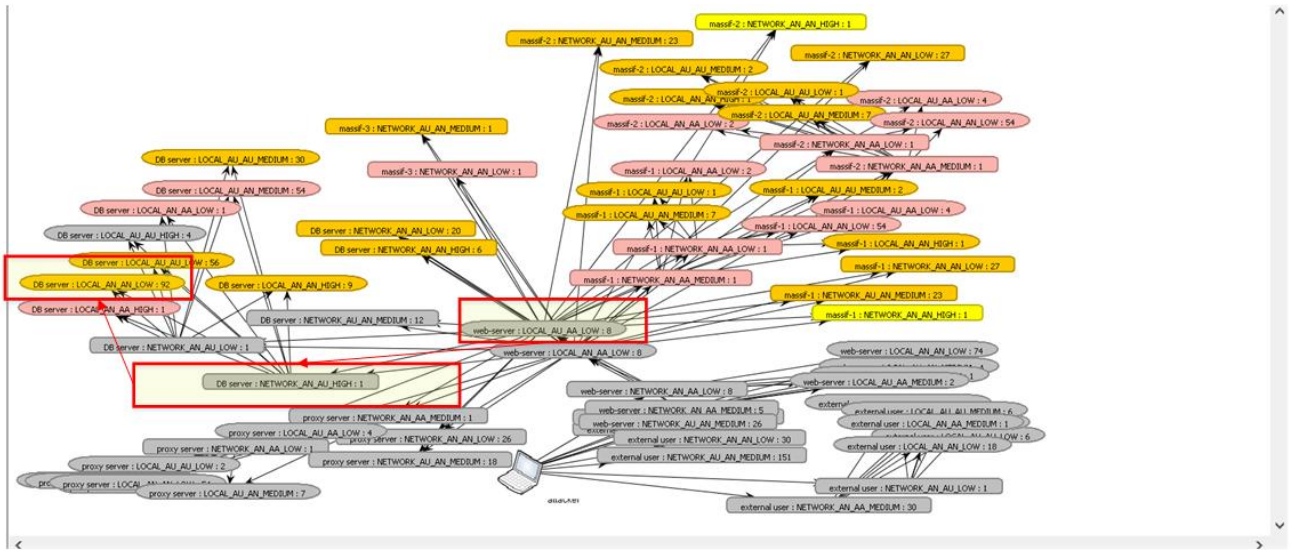


Рисунок 33 – Фрагмент интерфейса СОЗВК с графом атак после поступления последовательности событий безопасности

При получении события безопасности можно выделить ряд узлов, в направлении которых предположительно будет развиваться атака (по повышению уровня риска). После первого события выделяется достаточно большое количество узлов, которое снижается при обработке новых событий, что позволяет уже после второго события

локализовать целевой хост. Таким образом, изменение уровня риска позволяет локализовать цель атаки. На графике (рисунок 34) приведены изменения значений риска для обрабатываемых узлов графа после поступления первого (синяя кривая), второго (красная кривая) и третьего (зеленая кривая) события. Подробный анализ изменений значений риска для узлов графа приведен в приложении О.

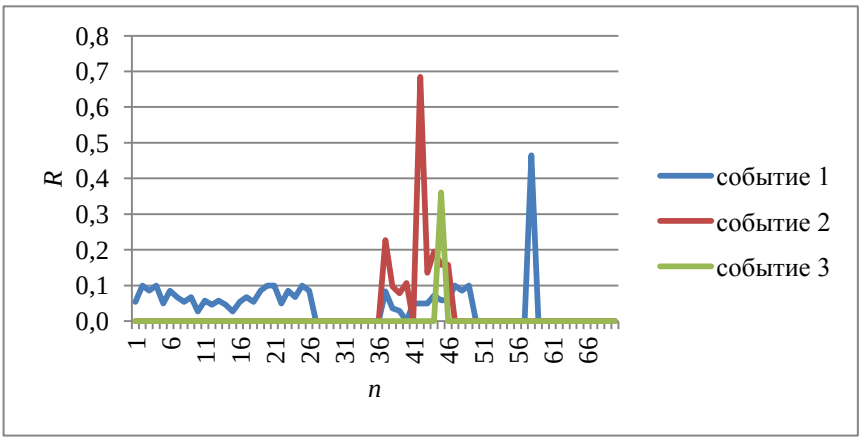


Рисунок 34 – График изменения значений риска *R* для узлов графа *n* после поступления последовательности событий безопасности

Процесс выбора контрмер для данной последовательности начинается уже после поступления первого события, так как для узлов, которые предположительно могут быть скомпрометированы на следующем шаге риск выше порогового значения («Высокий»).

При проведении экспериментов использовался следующий тестовый стенд: ПК на базе процессора Intel Core i7 и 8 GB ОЗУ. Для проведения экспериментов использовался набор контрмер, предложенный в [105]. Контрмеры выбраны для угроз категории «несанкционированный доступ к компьютерам, данным, сервисам и приложениям». Список контрмер, средств их реализации и соответствующие оценки стоимости и эффективности сформированы экспертами и приведены в таблице 10 (в соответствии с полями модели контрмеры, приведенными в главе 2). Контрмеры выбраны в соответствии с таблицей 7 для выбранной категории угроз. В примере рассматриваются следующие контрмеры для статического режима: ничего; межсетевой экран (категория «управление безопасностью сети» по классификации [10]); СОВ-СПВ (категории «обнаружение и предотвращение вторжений» и «управление безопасностью сети»); средства многофакторной аутентификации (категория «идентификация и

аутентификация»); средства мониторинга сети (категория «управление безопасностью сети»). Изначально в состав сети 1 добавлен один межсетевой экран и SIEM-система.

Таблица 10 – Контрмеры и средства их реализации [105]

Название контрмеры (меры динамического режима)	Стоимость контрмеры, в год (показатель)	Средство реализации (или меры статического режима)	Стоимость средства, в год (показатель)	Область действия средства	Эффективность контрмеры (показатель)	Тип влияния на граф атак	Область действия
Ничего не делать	0	–	0	–	0	–	–
Блокировка подозрительной учетной записи	0	Межсетевой экран	2000 €	Подграф Подсеть	[1 1 1]	Удаление связи	Граф атак/граф зависимостей сервисов
Активация СОВ в стратегических местах	600 €	СОВ-СПВ или SIEM	2500 €	Подграф Подсеть	[0,7 0,7 0,7]	Изменение связи	Граф атак
Смена портов соединения	400 €	СОВ-СПВ	2500 €	Подграф Подсеть	[0,55 0,55 0,55]	Изменение связи	Граф атак
Активация многофакторной аутентификации	1200 €	Средства многофактор-ной аутентифика-ции	1800 €	Хост Узел графа	[0,85 0,85 0,85]	Изменение связи	Граф атак
Активация правил аномального поведения	650 €	Средства мониторинга сети	2000 €	Сеть Граф	[0,7 0,7 0,7]	Изменение связи	Граф атак
Временная блокировка подозрительной учетной записи	0	Межсетевой экран	2000 €	Подграф Подсеть	[1 1 1]	Удаление связи	Граф атак/граф зависимостей сервисов
Управление уязвимостями/об новлениями	1200 €	–	–	Узел графа	[1 1 1]	Удаление связи	Граф атак

Контрмеры для динамического режима: ничего; блокировка подозрительной учетной записи; активация СОВ в стратегически важных местах (обнаружение и избегание нарушителей в ключевых точках инфраструктуры); смена портов соединения (позволяет избежать быстрого распознавания сети); активация многофакторной аутентификации (дополнительная аутентификация в виде дополнительных вопросов или биометрики); активация правил аномального поведения (наложение более строгих правил на поведение пользователей); временная блокировка подозрительной учетной записи (на 24, 48 или 72 часа). Контрмеры для динамического режима, соответствующие средствам реализации, добавленным в сеть 1, выделены в таблице 10 серым цветом.

На рисунке 35 представлен фрагмент листинга вывода контрмер в разработанном программном средстве для сети 1. Для рассматриваемой последовательности атаки результаты выбора контрмер после первого события: “Block connection” (блокировка подозрительной учетной записи) к хостам massif-1, massif-2 и massif-3 с помощью брандмауэра massif-i; для узлов хоста DB-сервер предлагается обновить ПО для удаления уязвимостей (“Patch vulnerable application”) (листинг вывода результата представлен на рисунке 36).

Countermeasures for host massif-1:
Possible action: ID=50 type=CounterMeasure name=Block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Possible action: ID=51 type=CounterMeasure name=Temporal block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Countermeasure: ID=53 type=CounterMeasure name=No countermeasures
Possible action: ID=52 type=CounterMeasure name=Patch vulnerable application
Countermeasures for host massif-2:
Possible action: ID=50 type=CounterMeasure name=Block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Possible action: ID=51 type=CounterMeasure name=Temporal block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Countermeasure: ID=53 type=CounterMeasure name=No countermeasures
Possible action: ID=52 type=CounterMeasure name=Patch vulnerable application
Countermeasures for host massif-3:
Possible action: ID=50 type=CounterMeasure name=Block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Possible action: ID=51 type=CounterMeasure name=Temporal block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Countermeasure: ID=53 type=CounterMeasure name=No countermeasures
Possible action: ID=52 type=CounterMeasure name=Patch vulnerable application
Countermeasures for host massif-i:
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Actions: [ID=50 type=CounterMeasure name=Block connection, ID=51 type=CounterMeasure name=Temporal block connection, ID=53 type=CounterMeasure name=No countermeasures]
Possible action: ID=50 type=CounterMeasure name=Block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Possible action: ID=51 type=CounterMeasure name=Temporal block connection
Implementation tool: ID=49 type=CounterMeasure name=Firewall
Countermeasure: ID=53 type=CounterMeasure name=No countermeasures
Possible action: ID=52 type=CounterMeasure name=Patch vulnerable application
Countermeasures for host attacker:
Possible action: ID=52 type=CounterMeasure name=Patch vulnerable application
Countermeasures for host external user:
Possible action: ID=52 type=CounterMeasure name=Patch vulnerable application
Countermeasures for host web-server:
Possible action: ID=52 type=CounterMeasure name=Patch vulnerable application
Countermeasures for host DB server:

Рисунок 35 – Результаты вывода контрмер для сети 1

После второго события локализовано, что атака идет на DB-сервер и предлагается контрмера “Patch vulnerable application” для его узлов (нет необходимости реализовывать контрмеру “Block connection”). Тем не менее, представляется целесообразным применение мер уже после первого события, так как уровень риска узлов подграфа превышает «Высокий» и ожидание новых событий может привести к компрометации критических узлов в случае выбора атакующим другого пути атаки.

Selected countermeasures: Countermeasure ID=50 type=CounterMeasure name=Block connection. Implementation tool: ID=49 type=CounterMeasure name=Firewall, host massif-I, should be implemented for nodes: [massif-1; massif-2; massif-3]
Countermeasure ID=52 type=CounterMeasure name=Patch vulnerable application should be implemented for nodes: [DB-server]

Рисунок 36 – Результаты выбора контрмер для сети 1 после поступления события безопасности 1

Отметим, что при экспериментах целевые узлы попадали в группу потенциальных целей уже после обработки первого события, что позволяло выбирать эффективные

контрмеры на уровне подсети. Ожидание новых событий целесообразно при достаточном расстоянии на графе до критических узлов сети (что обеспечивается низким уровнем риска в данном случае), и позволяет снизить затраты на реализацию контрмер без потерь.

Для каждой спецификации были проведены эксперименты по точности локализации пути атаки (показатель *индекс отклонения сценариев атак*и требований, поставленных в главе 1). Индекс отклонения сценариев атаки *AC* измерялся следующим образом: $AC=(\Delta Pr_{max}-\Delta Pr_r)/(\Delta Pr_r-\Delta Pr_{min})$ где ΔPr_{max} – максимальное суммарное изменение вероятности атаки после фиксации события безопасности для узлов выявленного сценария атаки, ΔPr_{min} – минимальное суммарное изменение вероятности атаки после фиксации события безопасности для узлов выявленного сценария атаки, ΔPr_r – суммарное изменение вероятности атаки для узлов реального пути атаки. Ограничения: $\Delta Pr_r \neq \Delta Pr_{min}$ (в этом случае реальная последовательность совпадает с наименее вероятной).

Значение ΔPr_{max} определялось следующим образом: $\Delta Pr_{\max} = \sum_{i=1}^{n} \Delta CPr_i + \sum_{i=1}^{m} \Delta PPr_i$, где n – длина фрагмента сценария атаки после инцидента (количество узлов графа атак), ΔCPr_i – максимальное изменение вероятности компрометации прямых потомков для i-го узла потомка инцидента: $\Delta CPr_i = \max_{l=1}^{k} \Delta Pr_l$, где k – количество прямых потомков i-го узла потомка инцидента; m – длина фрагмента сценария атаки до инцидента, ΔPPr_i – максимальное изменение вероятности компрометации прямых узлов предков для узла инцидента: $\Delta PPr_i = \max_{l=1}^{k} \Delta Pr_l$, где k – количество прямых предков i-го узла предка инцидента.

Значение ΔPr_{min} определялось следующим образом: $\Delta Pr_{min} = \sum_{i=1}^{n} \Delta PPr_i + \sum_{i=1}^{m} \Delta CPr_i$, где n – длина фрагмента сценария атаки после инцидента (количество узлов графа атак), ΔCPr_i – минимальное изменение вероятности компрометации прямых потомков для i-го узла потомка инцидента: $\Delta CPr_i = \max_{l=1}^{k} \Delta Pr_l$, где k – количество прямых потомков i-го узла потомка инцидента; m – длина фрагмента сценария атаки до инцидента, ΔPPr_i – минимальное изменение вероятности компрометации прямых узлов предков для узла инцидента: $\Delta PPr_i = \max_{l=1}^{k} \Delta Pr_l$, где k – количество прямых предков i-го узла предка инцидента.

Значение ΔPr_r определялось как сумма значений изменения вероятности атаки для узлов реального пути атаки.

Отметим, что чем меньше полученное отклонение, тем точнее локализован путь.

Для определения точности локализации пути атаки также вычислялась дисперсия δ^2, позволяющая определить разброс значений изменения вероятности. Чем больше данное значение, тем больше изменение вероятности спрогнозированного пути отличается от изменения вероятностей других возможных путей атаки. Дисперсия δ^2 рассчитывалась по формуле: $\delta^2 = \frac{1}{n} \times \sum_{i=1}^{n} (AC_i - \overline{AC})^2$, где n – количество последовательностей атаки; AC_i – отклонение i-й последовательности; $\overline{AC}$ – среднее отклонение.

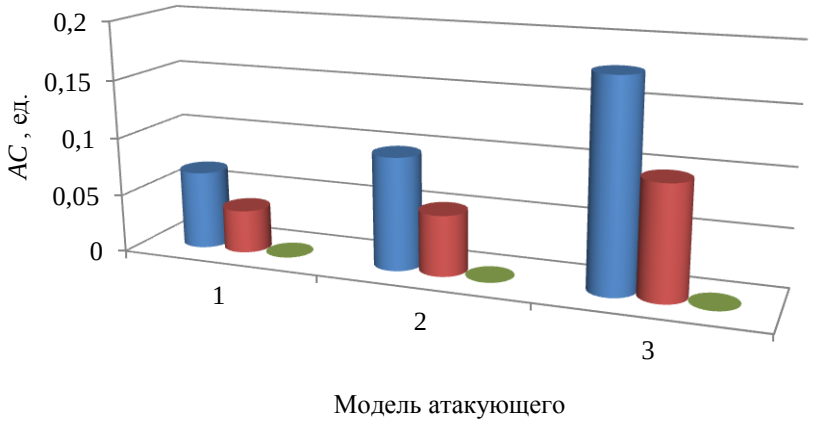
Для каждой сети для каждого типа атакующего было сгенерировано десять различных последовательностей атак. За результат AC было взято среднее по десяти экспериментам. Дисперсия δ^2 рассчитывалась для каждых десяти последовательностей. Результаты экспериментов сведены в таблицу 11.

Таблица 11 – Результаты экспериментов по точности локализации пути атаки

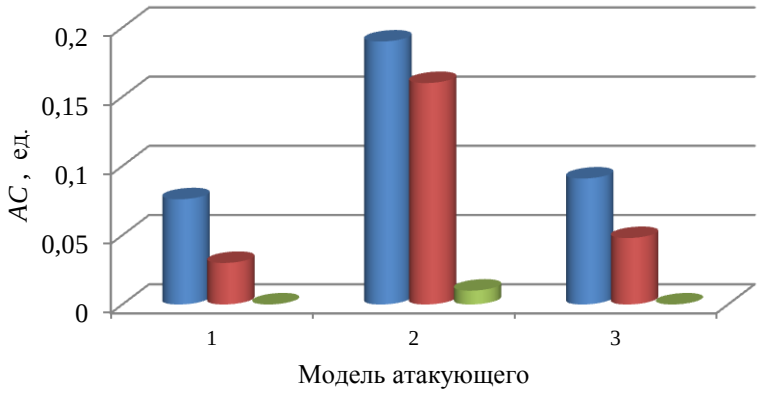
Модель сети	Сеть 1			Сеть 2			Сеть 3		
Событие/ модель атакующего	Атакующий 1	Атакующий 2	Атакующий 3	Атакующий 1	Атакующий 2	Атакующий 3	Атакующий 1	Атакующий 2	Атакующий 3
Событие 1	AC=0,067 $\delta^2 = 4{,}3 \cdot 10^{-4}$	AC=0,097 $\delta^2 = 3{,}8 \cdot 10^{-4}$	AC=0,177 $\delta^2 = 1{,}4 \cdot 10^{-3}$	AC=0,105 $\delta^2 = 2{,}1 \cdot 10^{-4}$	AC=0,024 $\delta^2 = 1{,}96 \cdot 10^{-4}$	AC=0,091 $\delta^2 = 2{,}1 \cdot 10^{-4}$	AC=0,2 $\delta^2 = 8 \cdot 10^{-5}$	AC=0,24 $\delta^2 = 8 \cdot 10^{-5}$	AC=0,29 $\delta^2 = 8 \cdot 10^{-5}$
Событие 2	AC=0,037 $\delta^2 = 2{,}6 \cdot 10^{-4}$	AC=0,052 $\delta^2 = 2{,}3 \cdot 10^{-5}$	AC=0,097 $\delta^2 = 6 \cdot 10^{-5}$	AC=0,08 $\delta^2 = 2 \cdot 10^{-4}$	AC=0 $\delta^2 = 1{,}7 \cdot 10^{-6}$	AC=0,048 $\delta^2 = 1{,}3 \cdot 10^{-4}$	AC=0,038 $\delta^2 = 9 \cdot 10^{-7}$	AC=0,022 $\delta^2 = 2 \cdot 10^{-7}$	AC=0,063 $\delta^2 = 8 \cdot 10^{-7}$
Событие 3	AC=0 $\delta^2 = 1{,}6 \cdot 10^{-7}$	AC=0,0 $\delta^2 = 1{,}7 \cdot 10^{-7}$	AC=0 $\delta^2 = 2 \cdot 10^{-6}$	AC=0,196 $\delta^2 = 1{,}3 \cdot 10^{-6}$	AC=0 $\delta^2 = 4{,}8 \cdot 10^{-8}$	AC=0 $\delta^2 = 1{,}6 \cdot 10^{-8}$	AC=0 $\delta^2 = 5 \cdot 10^{-9}$	AC=0 $\delta^2 = 5 \cdot 10^{-9}$	AC=0 $\delta^2 = 6 \cdot 10^{-9}$

Значения отклонения для сети 1, 2 и 3 представлены на рисунке 37: для каждой модели атакующего первый столбец показывает уровень отклонения после события 1, второй – после события 2, и третий – после события 3. Отметим, что индекс уменьшается с увеличением количества обработанных событий, и для выбранных

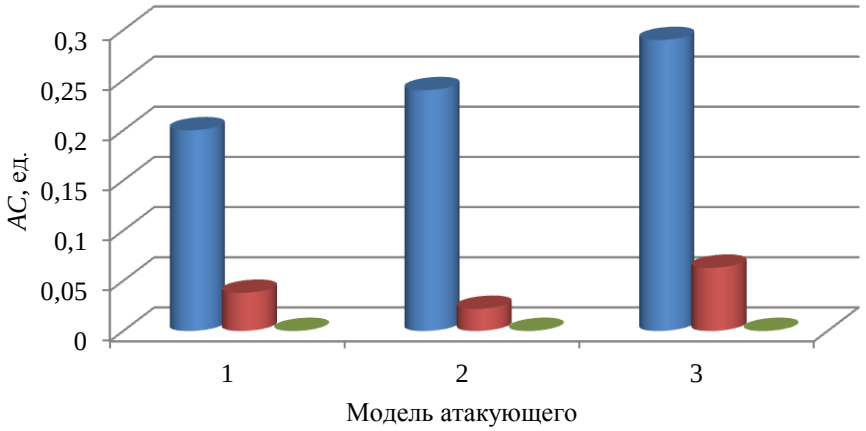
спецификаций сетей уже после третьего события приближается к нулю. Таким образом, требование, поставленное в главе 1: $AC \to \min$, удовлетворяется.



а)



б)



в)

Рисунок 37 – Отклонение от реального пути атаки для атакующего 1, 2 и 3 после поступления событий для а) сети 1; б) сети 2; в) сети 3.

Для каждой сети были проведены эксперименты по определению выигрыша в результате реализации контрмер (показатель *выигрыш в случае реализации контрмер* требований, поставленных в главе 1). Данный показатель определяется на основе индекса *AL*. Индекс *AL* определяется следующим образом: $AL = EL - Losses_{after}$, где *EL* – потери для КС для последовательности атаки в случае, если никаких контрмер предпринято не будет; $Losses_{after}$ – потери в случае, если контрмеры реализованы.

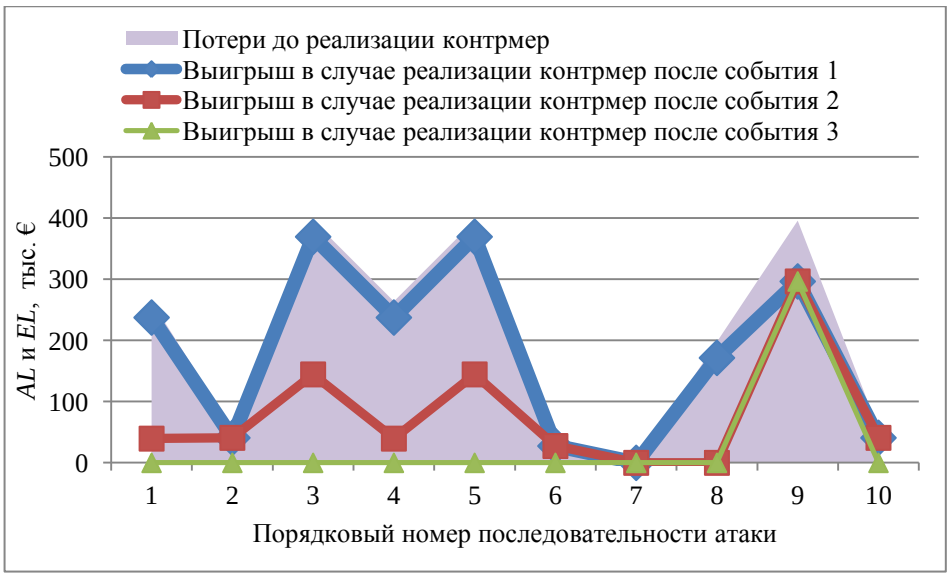
На рисунке 38, рисунке 39 и рисунке 40 приведены результаты экспериментов для сети 1, сети 2 и сети 3, соответственно (для атакующего 1, атакующего 2 и атакующего 3). Поверхность обозначает ожидаемые потери до реализации контрмер. Самая широкая линия обозначает выигрыш в случае реализации контрмер после первого события, средняя по ширине линия обозначает выигрыш в случае реализации контрмер после первого события, самая тонкая линия обозначает выигрыш в случае реализации контрмер после первого события.

Из графиков на рисунке 38, рисунке 39 и рисунке 40 видно, что наибольший выигрыш достигается в случае реализации контрмер после поступления события 1. Это связано с тем, что уровень риска превышает пороговое значение уже после события 1, в случае, если в этот момент не реализовать контрмер, атакующий успевает нанести ущерб системе и выигрыш при реализации контрмер после события 2 уже значительно ниже.

Значения показателей *EL*, $Losses_{after}$ и *AL*, полученные при экспериментах, приведены в приложении П (таблица П.1).

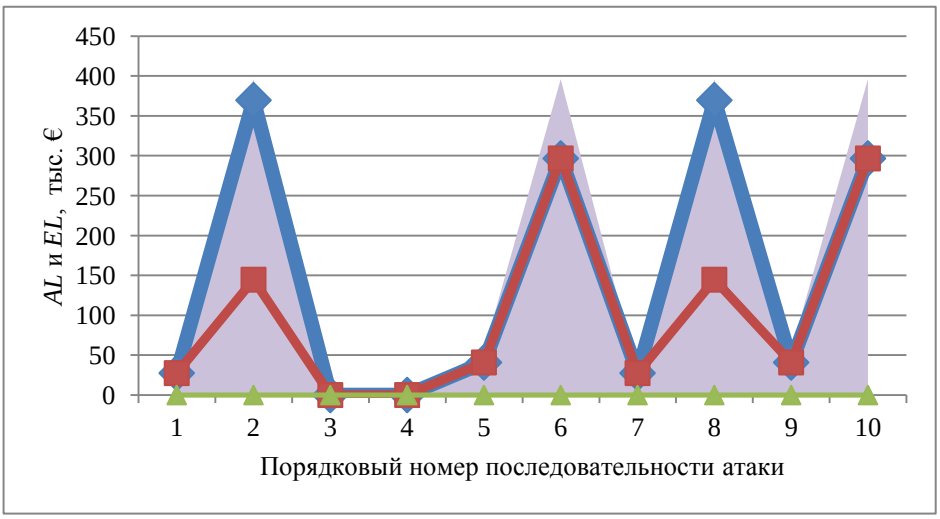
В случае своевременной реализации контрмер выигрыш стремится к потерям, ожидаемым в случае успешной реализации атаки. Таким образом, реализация контрмер позволяет снизить потери в результате успешной реализации атаки и требование, поставленное в главе 1: $AL \to EL$, удовлетворяется.

Риск компрометации ценных активов напрямую зависит от ожидаемых потерь, таким образом, применение разработанных методик позволяет снизить уровень риска и целевая функция, заданная в главе 1: $RiskCalc(a, C) \to \min$ (где *C* – множество средств защиты и защитных мер, рекомендуемых системой оценки защищенности и выбора защитных мер для атаки *a*), удовлетворяется.



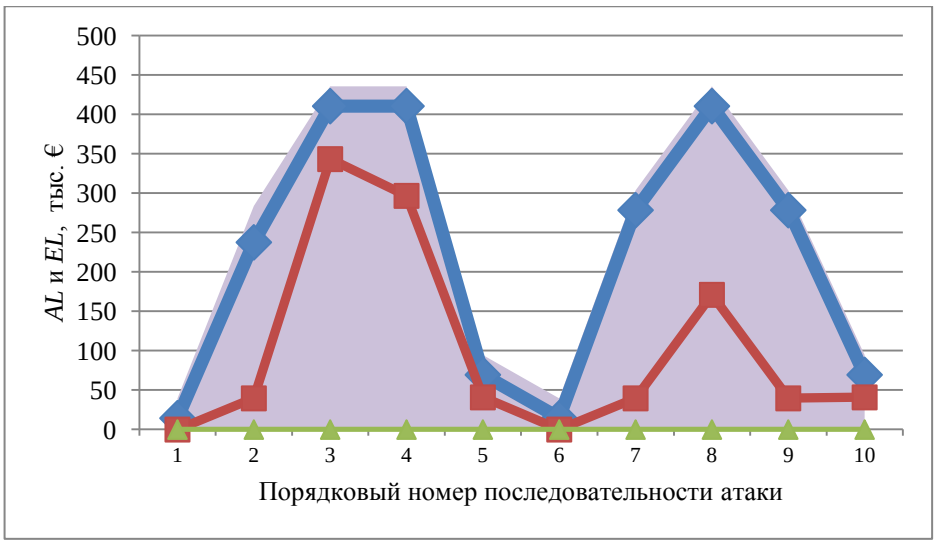
а)

б)

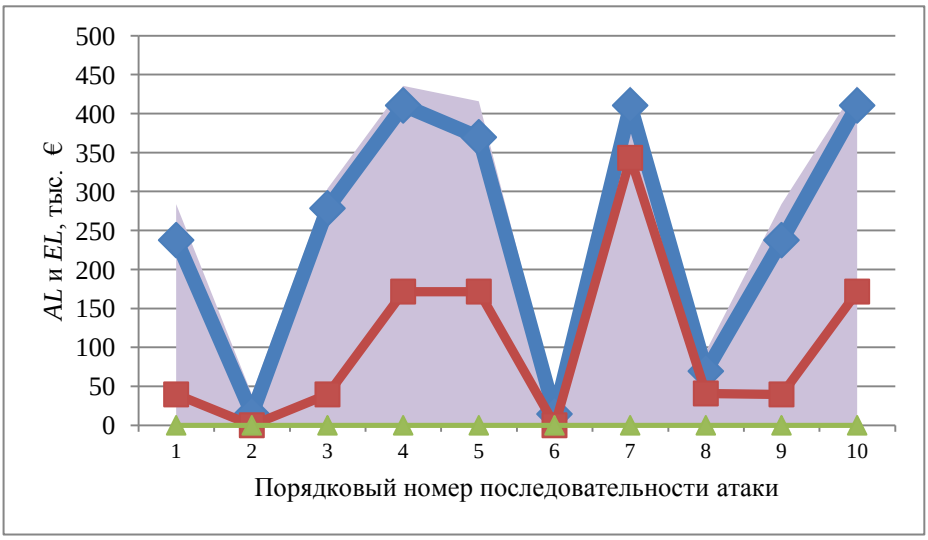


в)

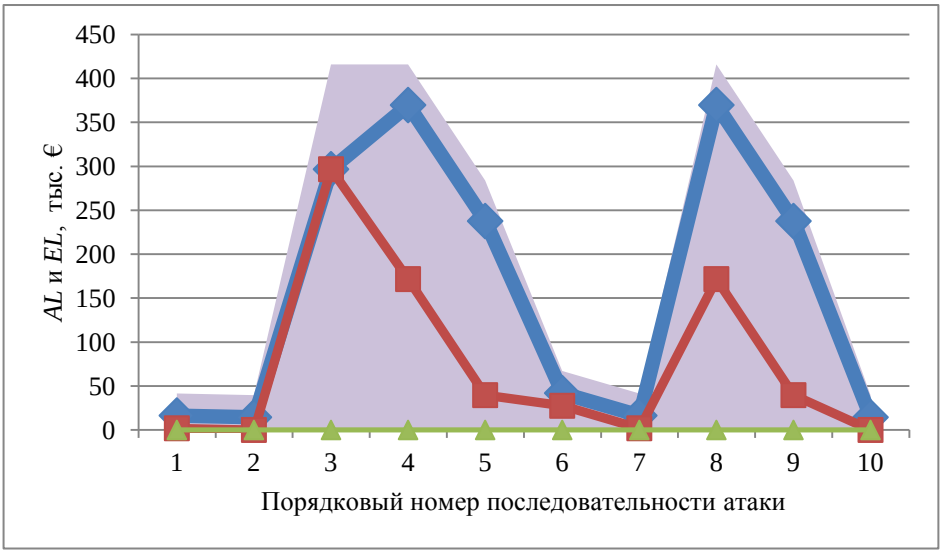
Рисунок 38 – Уровень потерь до и выигрыш после реализации контрмер для сети 1 для а) атакующего 1; б) атакующего 2; в) атакующего 3



а)

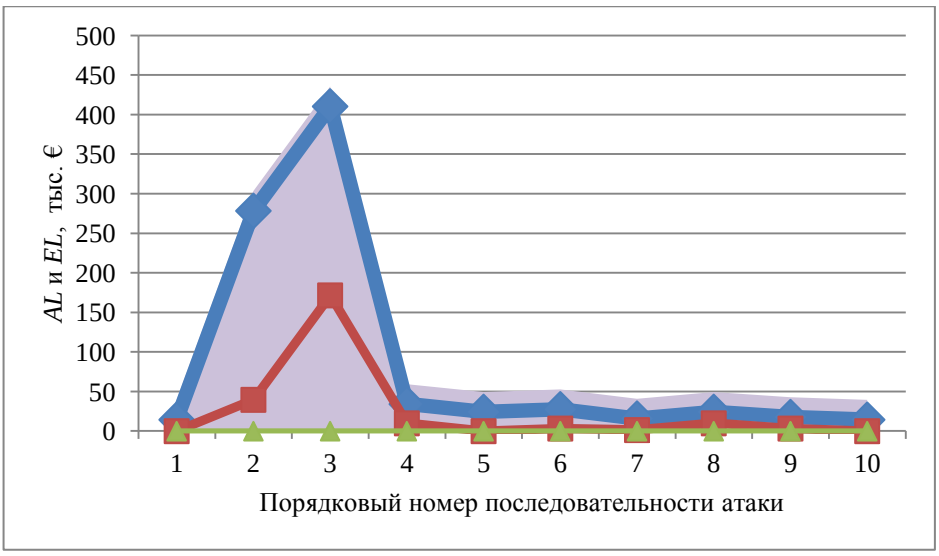


б)

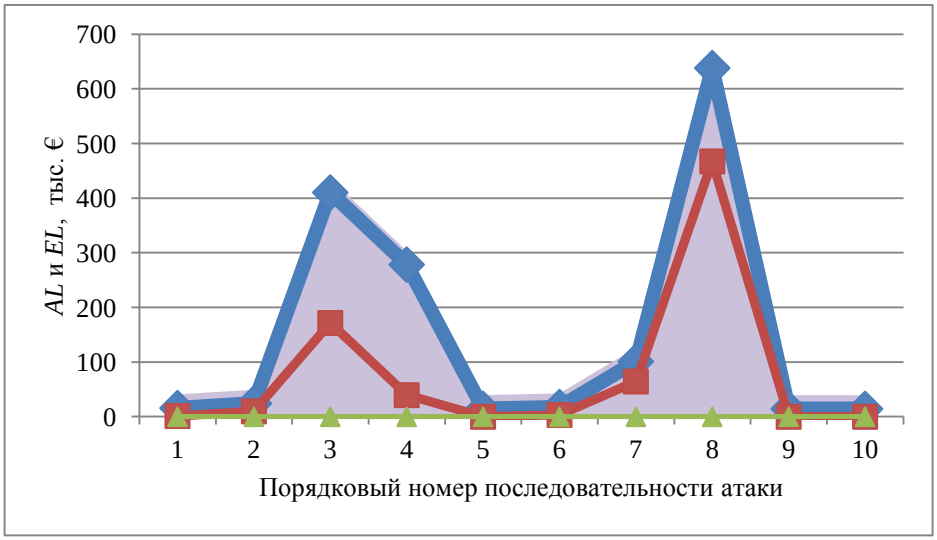


в)

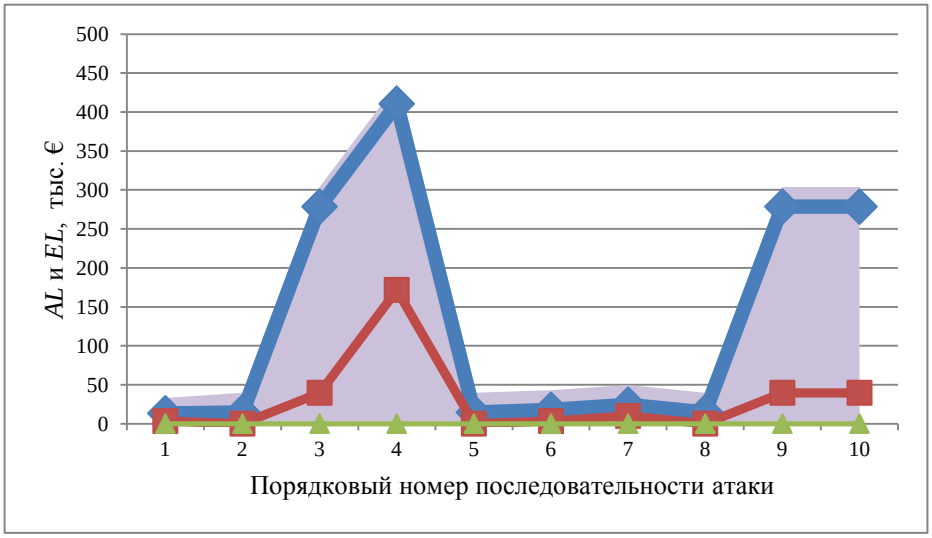
Рисунок 39 – Уровень потерь до и выигрыш после реализации контрмер для сети 2 для а) атакующего 1; б) атакующего 2; в) атакующего 3



а)



б)



в)

Рисунок 40 – Уровень потерь до и выигрыш после реализации контрмер для сети 3 для а) атакующего 1; б) атакующего 2; в) атакующего 3

Для подтверждения эффективности предлагаемых методик, рассмотрим функциональные и нефункциональные требования, поставленные в главе 1. Список функциональных требований, показателей для оценки их выполнимости и способов оценки приведен в таблице 12. Эксперименты показали, что поставленные функциональные требования выполняются.

Таблица 12 – Функциональные требования

Наименование требования	Описание показателя	Способ оценки
Получение адекватной и актуальной оценки защищенности КС на основе доступных входных данных в статическом и динамическом режимах работы	Реализация базового функционала компонента оценки защищенности	Функциональное тестирование
Учет характеристик атакующего	Реализация базового функционала компонента оценки защищенности	Функциональное тестирование
Учет взаимосвязей между сервисами КС	Реализация базового функционала компонента оценки защищенности	Функциональное тестирование
Учет стоимостных характеристик атак и защитных мер	Реализация базового функционала компонента оценки защищенности	Функциональное тестирование
Автоматизация процесса представления и обработки данных	Реализация базового функционала компонента обработки данных	Функциональное тестирование
Выбор наиболее адекватного решения по реагированию с учетом стоимостных требований в статическом и динамическом режимах работы	Реализация базового функционала компонента выбора контрмер	Функциональное тестирование
Учет событий безопасности, происходящих в КС, и переоценка ситуации по защищенности в соответствии с полученной информацией в динамическом режиме работы	Реализация базового функционала компонента оценки защищенности	Функциональное тестирование
Интеграция с SIEM-системами в динамическом режиме работы	Реализация базового функционала компонента обработки данных	Функциональное тестирование

В качестве нефункциональных были выделены требования к обоснованности, своевременности и ресурсопотреблению. Показателями обоснованности выступают: количество анализируемых сценариев атак; количество учитываемых параметров; точность выявления сценария атаки; выигрыш в случае реализации контрмер.

Количество анализируемых сценариев атак напрямую зависит от количества учитываемых уязвимостей. Предлагаемая методика учитывает все известные уязвимости программно-аппаратного обеспечения из открытых баз данных [119]. Таким образом, по данному параметру методика не уступает существующим аналогам.

При оценке защищенности применяются методики, учитывающие различные компоненты анализа защищенности. Предполагается, что комплексный учет характеристик различных компонентов позволит получить оценку, наиболее адекватно отражающую ситуацию по защищенности. В таблице 13 приводится сравнение

разработанной СОЗВК с существующими системами по учитываемым параметрам на качественном уровне.

Таблица 13 – Сравнение предложенной методики с существующими подходами

Подход	**Параметры, учитываемые при оценке**								
	Базовые показатели (множество дополнительных показателей, рассчитываемых на основе параметров КС)	Характеристики атаки (вероятность)	Характеристики атакующего (уровень навыков, положение, мотивация)	Ущерб от атаки (потери)	Общая оценка защищенности (уровень риска или уровень защищенности, поверхность атаки)	Затраты на контрмеры	Побочный ущерб от реализации контрмер	Эффективность контрмер	События безопасности
Poolsappasit и др. [125]	-	+ Байесовский граф атак	-	+- Устанавливается вручную	+ Ожидаемые потери	+- Устанавливается вручную	+- Устанавливается вручную	+	+
Dantu и др. [67]	-	+- Байесовская сеть доверия на основе профиля атакующего	+ Реалистичный профиль атакующего на основе набора характеристик	-	+ Вероятность атаки	-	-	+ Изменение характеристик атакующего, необходимых для компрометации узла	-
Kheir и др. [89]	-	-	-	+ На основе зависимостей сервисов	-	+- Устанавливается вручную	+ На основе зависимостей сервисов	+ На основе зависимостей сервисов	-
Wu и др. [145]	-	+- Граф атак на основе достижимости сервиса	-	+ На основе зависимостей сервисов и исторических данных об инцидентах	-	-	+ Результаты атаки после реализации контрмер	+ Результаты атаки после реализации контрмер	+
Степашкин [37]	-	+ Граф атак	+- Позиция атакующего и привилегии в сети	+- Качественные оценки критичности ресурсов	+ Ущерб от атаки и вероятность атаки	-	-	-	-
Jahnke и др. [85]	-	+- Граф доступности	-	+- Изменение доступности ресурса	-	-	+- Изменение доступности ресурса	+- Изменение доступности ресурса	+- Устанавливается вручную
Lippmann и др. [98]	-	+ Граф атак	-	+- Количество хостов для которых получен доступ администратора	-	-	-	+- Снижение количества скомпрометированных хостов	-
Granadillo и др. [74]	-	+ Поверхность атаки	-	+ Затронутый ресурс системы	+ Поверхность атаки	-	+ Объем ресурсов, не затронутых атакой	+ Снижение объема атаки	-
Разработанная методика	+	+ Байесовский граф атак	+- Позиция атакующего и привилегии в сети, переопределяется на основе событий	+ На основе зависимостей сервисов	+ Ущерб от атаки и вероятность атаки	+- Устанавливается вручную	+ На основе зависимостей сервисов	+- Устанавливается вручную	+

Предполагается, что качество оценки зависит от полноты учитываемых входных данных. В таблице используются обозначения: «+» – система полностью поддерживает данный тип параметров; «-» – поддержка данного типа отсутствует; «+-» – система поддерживает данный тип входных данных только частично. Так, например, в [125] при оценке защищенности и выборе контрмер в качестве входных данных учитываются характеристики атаки, а именно, вероятность атаки, которая рассчитывается на основе Байесовского графа атак, тем не менее, такие параметры как ущерб от атаки, затраты на контрмеры и побочный ущерб от реализации контрмер, хотя и учитываются, устанавливаются вручную экспертами. В [89], напротив, при оценке защищенности и выборе контрмер не учитывается параметр вероятности атаки, а ущерб от атаки, затраты на контрмеры и побочный ущерб от реализации контрмер рассчитываются на основе графов зависимостей сервисов. Из таблицы видно, что разработанная система учитывает большее количество параметров, чем существующие системы, из чего можно сделать вывод о ее превосходстве над аналогами.

Предполагается, что точность выявления сценария атаки, а также выигрыш в случае реализации контрмер зависит от количества учитываемых параметров (превосходство разработанной системы над существующими аналогами по количеству учитываемых параметров показано выше) и качества анализа. Качество анализа было показано на экспериментах по определению соответствия отклонения выявленного сценария атаки от реального (рисунок 37) и выигрыша в случае реализации контрмер (рисунок 38, рисунок 39, рисунок 40) заявленным в главе 1 требованиям.

Таким образом, можно сделать вывод о том, что значение свойств *обоснованности* не уступает существующим методикам и удовлетворяет требованиям.

Для оценки *своевременности* измерим время, необходимое для выполнения основных этапов методик, в том числе в статическом режиме: (1) вычисление показателей на основе графа зависимостей сервисов; (2) построение байесовского графа атак; (3) вычисление показателей на основе байесовского графа атак; (4) вычисление интегрального показателя риска (оценка уровня защищенности). В динамическом режиме: (1) отображение события на граф атак; (2) пересчет показателей на основе байесовского графа атак; (3) выбор контрмер.

В общем случае, время выполнения методики оценки защищенности КС и выбора контрмер будет складываться из продолжительности времени выполнения вышеперечисленных этапов:

$$TIME^{C} = T_1^{C} + T_2^{C} + T_3^{C} + T_4^{C},$$

$$TIME^{Д} = T_1^{Д} + T_2^{Д} + T_3^{Д},$$

где T_i – время выполнения i-го этапа, $i \in [1;4]$.

Время выполнения рассматривается как случайная величина, вероятность которой подчиняется нормальному закону распределения [41]. На практике, для оценки времени выполнения обычно используют закон бета-распределения в интервале $[t_{\min}, t_{\max}]$ с плотностью распределения [31]:

$$f(t) = \begin{cases} \dfrac{(t - t_{\min})^{\alpha-1}(t_{\max} - t)^{\beta-1}}{(t_{\max} - t_{\min})^{\alpha+\beta-1} B(\alpha,\beta)}, t_{\min} \le t \le t_{\max} \\ 0, t_{\max} \le t \le t_{\min} \end{cases},$$

где t_{min} и t_{max}, соответственно, минимальное и максимальное время выполнения; t – случайная величина, определяющая время выполнения; $B(\alpha,\beta)$ – функция Эйлера; $\alpha > 0$, $\beta > 0$ – параметры бета-распределения.

Для вычисления ожидаемого времени выполнения методик статического и динамического режимов и дисперсии воспользуемся двухоценочной методикой, при которой область определения времени задается минимальной T_{min} и максимальной T_{max} оценками [41]:

$$T_i = \frac{3T_i^{\min} + 2T_i^{\max}}{5},\ \sigma^2(T_i) = 0,4 \cdot \left(T_i^{\max} - T_i^{\min}\right)^2.$$

Вероятность того, что суммарное время выполнения методик не превысит допустимого значения $T^{ДОП}$, вычисляется по формуле [36]:

$$P_{СВ}\left(T \le T^{ДОП}\right) = \Phi(Z),$$

где $\Phi(Z)$ – значение функции Лапласа [39] при $Z = \dfrac{T^{ДОП} - \sum_{i=1}^{n} T_i}{\sqrt{\sum_{i=1}^{n} \sigma_i^2(T_i)}}$.

Результаты экспериментов по оценке своевременности разработанных методик для модели КС, содержащей 500 хостов, представлены в таблицах 14 и 15.

Таблица 14 – Временные показатели методик оценки защищенности в статическом режиме работы

Этап	$T_i^{\min}$, сек	$T_i^{\max}$, сек	$T_i = \frac{3T_i^{\min} + 2T_i^{\max}}{5}$	$\sigma^2(T_i) = 0,4 \cdot \left(T_i^{\max} - T_i^{\min}\right)^2$
1	28,6	36,5	31,76	24,96
2	0,6	1,59	0,996	0,39
3	3,27	8,45	5,342	6,99
4	0,015	0,032	0,02	0,0001
Итого в статическом режиме, сек			**38,118**	**32,34**

Тогда, учитывая значения функции Лапласа, заданные таблично [39], значение функции Лапласа Ф(Z) для $TIME^{ДОП}$=60 секунд в статическом режиме работы:

$$\Phi\left(\frac{TIME^{ДОП} - \sum_{i=1}^{4} T_i}{\sqrt{\sum_{i=1}^{4} \sigma_i^2(T_i)}}\right) = \left(\frac{60 - 38,118}{\sqrt{32,34}}\right) \approx 3,86.$$

В динамическом режиме работы значение функции Лапласа $\Phi(Z)$ при $TIME^{ДОП} = 10$ секунд:

$$\Phi\left(\frac{TIME^{ДОП} - \sum_{i=1}^{4} T_i}{\sqrt{\sum_{i=1}^{4} \sigma_i^2(T_i)}}\right) = \left(\frac{10 - 3,61}{\sqrt{4,05}}\right) \approx 3,18.$$

Таблица 15 – Временные показатели методик оценки защищенности и выбора контрмер в динамическом режиме работы

Этап	$T_i^{\min}$, сек	$T_i^{\max}$, сек	$T_i = \frac{3T_i^{\min} + 2T_i^{\max}}{5}$	$\sigma^2(T_i) = 0,4 \cdot \left(T_i^{\max} - T_i^{\min}\right)^2$
1	0,016	0,141	0,066	0,006
2	2,27	5,45	3,542	4,044
3	0,001	0,015	0,0066	0,00007
Итого в динамическом режиме, сек			**3,61**	**4,05**

Тогда, для КС из 500 хостов, вероятность выполнения методик за заданное время: $P_{СВ}\left(TIME_N \le TIME^{ДОП}\right) = 0,99989$ в статическом режиме работы, и

$P_{СВ}\left(TIME_N \le TIME^{ДОП}\right) = 0,99853$ в динамическом режиме работы. Таким образом, требования, поставленные в главе 1 к *своевременности* ($P_{СВ}^{ДОП} = 0,99$), удовлетворены.

Ресурсопотребление характеризует номенклатуру и количество необходимых программных и аппаратных средств, объемы необходимых информационных массивов, кадровые и другие ресурсы, используемые при работе СОЗВК. Оценка ресурсопотребления проводилась по ряду показателей [37, 43]: (1) доля загруженности центрального процессорного устройства $R_{ЦП} = \frac{Q_{ЦП}^{C}}{Q_{ЦП}^{ОБЩ}}$, где $Q_{ЦП}^{C}$ – время центрального процессора, затраченное на выполнение оценки защищенности и выбор контрмер СОЗВК, $Q_{ЦП}^{ОБЩ}$ – общее процессорное время; (2) использование сетевого ресурса $R_{СЕТЬ} = \frac{Q_{СЕТЬ}^{C}}{Q_{СЕТЬ}^{ОБЩ}}$, где $Q_{СЕТЬ}^{C}$ – объем передаваемых по сети данных при работе СОЗВК, $Q_{СЕТЬ}^{ОБЩ}$ – общий сетевой ресурс; (3) использование жесткого диска $R_{ЖД} = \frac{Q_{ЖД}^{C}}{Q_{ЖД}^{ОБЩ}}$, где $Q_{ЖД}^{C}$ – объем памяти на жестком диске, используемый при работе СОЗВК, $Q_{ЖД}^{ОБЩ}$ – общий объем жесткого диска; (4) использование оперативной памяти $R_{ОП} = \frac{Q_{ОП}^{C}}{Q_{ОП}^{ОБЩ}}$, где $Q_{ОП}^{C}$ – объем оперативной памяти, используемый при работе СОЗВК, $Q_{ОП}^{ОБЩ}$ – общий объем оперативной памяти.

Для соответствия требованиям все вышеперечисленные показатели r должны удовлетворять условию $r \le R^{ДОП}$, где $R^{ДОП}$=0,75. Значение $R^{ДОП}$ определено экспертным путем, с учетом того, что для выполнения задач оценки и выбора контрмер предполагается выделение отдельного компьютера, 25% ресурсов выделяется на общие задачи ОС и сопутствующих программ.

Для проведения экспериментов был выбран стандартный для разработчика ПО персональный компьютер со следующими параметрами: (1) центральный процессор – Intel Core i7-4510U CPU @2.00GHz (4 ядра); (2) максимальная пропускная способность сетевого канала – 100 Мб/с; (3) общий объем жесткого диска – 250 Гб; (4) общий объем оперативной памяти – 8 Гб.

По результатам экспериментов были получены показатели: (1) $R_{ЦП}$=0,25 (в процессе оценки защищенности и выбора контрмер было использовано на 100% только одно ядро процессора); (2) $R_{СЕТЬ} \le 0,039$ (сетевой канал используется во время

обновления базы данных уязвимостей и атак, а также для получения событий об изменениях в сети, размер пакета, содержащего описание события, не превышает 1 Кб, количество таких пакетов не более 40 в секунду, так как проверка изменений осуществляется раз в секунду, а количество хостов не превышает 40); $R_{ЖД} \le 0{,}00002$ (размер прототипа, реализующего предложенную методику, вместе со всеми необходимыми библиотеками, но без виртуальной машины Java, составляет около 5 Мб); $R_{ОП}$=0,275 (в процессе выполнения методики на платформе Windows 8, 64 бит, использовалось не более 1,1 Гб оперативной памяти).

По результатам оценки ресурсопотребления получаем: $P_{РЕС}\left(R \le R^{ДОП}\right) = 1.$ Таким образом, можно сделать вывод, что данный показатель удовлетворяет поставленным в главе 1 требованиям: $P_{РЕС}\left(r \le R^{ДОП}\right) \ge P_{РЕС}^{ДОП}$, где $P_{РЕС}^{ДОП} = 0{,}99$.

3.4 Предложения по использованию системы оценки защищенности и выбора контрмер

В настоящее время КС активно развиваются и применяются во многих критически важных коммерческих и государственных отраслях. Их дальнейшее развитие, а также важность повышения уровня защищенности таких систем, определены в стратегии развития информационного общества в РФ на 2014–2020 годы [38]. Разработанные методики оценки защищенности КС и выбора защитных мер позволят снизить уровень возможных потерь организаций в результате компьютерных атак за счет постоянного отслеживания и пересчета показателей защищенности в соответствии с поступающими данными о событиях в системе и своевременного применения адекватных контрмер. Это указывает на обширную область применения результатов данного исследования. Разработанная система может применяться как отдельный продукт для повышения уровня защищенности системы и выбора защитных мер, эффективных с точки зрения стоимости и снижения уровня риска. Для достижения этой цели система должна выполнять функции:

1) Получение адекватной и актуальной оценки защищенности КС на основе доступных входных данных в статическом и динамическом режимах работы. Для выполнения этой функции в рамках системы должен быть реализован компонент оценки защищенности.

2) Учет характеристик атакующего, в том числе, его целей, положения в сети, первичных знаний о сети, навыков и возможностей по реализации атак в статическом и динамическом режимах работы. Для выполнения этой функции компонент оценки защищенности должен учитывать модель атакующего.

3) Учет взаимосвязей между сервисами КС для тщательного учета возможного распространения ущерба в случае успешной реализации атак, или побочного ущерба при реализации защитных мер, в статическом и динамическом режимах работы. Для выполнения этой функции компонент оценки защищенности должен учитывать модель зависимостей сервисов, генерируемую функциями компонента обработки данных.

4) Учет стоимостных характеристик атак и защитных мер, для того, чтобы определить выигрыш в случае реализации защитных мер, в статическом и динамическом режимах работы. Для выполнения этой функции компонент оценки защищенности должен учитывать модель зависимостей сервисов и модель контрмер, генерируемые функциями компонента обработки данных.

5) Автоматизация процесса представления и обработки данных, применяемых для оценки защищенности и выбора защитных мер, в статическом и динамическом режимах работы. Для выполнения этой функции в рамках системы должен быть реализован компонент обработки данных.

6) Выбор наиболее адекватного решения по реагированию с учетом стоимостных требований в статическом и динамическом режимах работы. Для выполнения этой функции в рамках системы должен быть реализован компонент выбора контрмер.

7) Выявление слабых мест КС в статическом режиме работы. Для выполнения этой функции в рамках системы должен быть реализован компонент оценки защищенности.

8) Выявление возможных атак на КС и получение набора показателей защищенности, характеризующего их, в статическом режиме работы. Для выполнения этой функции в рамках системы должен быть реализован компонент оценки защищенности.

9) Выбор средств защиты для повышения уровня защищенности системы в статическом режиме работы. Для выполнения этой функции в рамках системы должен быть реализован компонент выбора контрмер.

10) Учет событий безопасности, происходящих в КС, и переоценка ситуации по защищенности в соответствии с полученной информацией в динамическом режиме работы. Для выполнения этой функции в рамках системы должен быть реализован компонент оценки защищенности.

11) Интеграция с SIEM-системами в динамическом режиме работы. Для выполнения этой функции в рамках системы должен быть реализован компонент обработки данных.

12) Выбор защитных мер для предотвращения развивающейся атаки в динамическом режиме работы. Для выполнения этой функции в рамках системы должен быть реализован компонент выбора контрмер.

Дополнительно необходимо разработать и подключить к системе внешние модули:

- *подсистема визуализации* для управления системой, ввода входных данных и вывода результатов работы системы;

- *база данных* для хранения конфигурации анализируемой КС.

Кроме того необходимо предусмотреть взаимодействие с сетевыми сканерами и сканерами уязвимостей, с системами обнаружения вторжений, SIEM-системой, или различными сетевыми устройствами, генерирующими события безопасности (в этом случае дополнительно необходимо разработать компонент обработки и анализа событий).

Для корректной работы системы рабочее место пользователя должно удовлетворять следующим требованиям: (1) ПК на базе процессора Intel Pentium 5 (или аналогичного): (а) объем оперативной памяти 4 ГБ; (б) объем жесткого диска 120 ГБ; (2) Java Runtime Environment версии 1.7.0 или выше.

Кроме того, данная система может применяться как основа компонента принятия решений активно развивающихся SIEM-систем. Применимость результатов исследования состоит в необходимости обработки генерируемых такими системами данных для формирования текущей картины по безопасности и выработки рекомендаций по реагированию. В данном случае необходимо разработать и протестировать компоненты преобразования выходных данных конкретной SIEM-системы во входные данные СОЗВК, и выходных данных СОЗВК во входные данные SIEM-системы. Также разработанная система может применяться как компонент СОВ и СПВ.

Для улучшения системы, разработанной в данном диссертационном исследовании, можно: (1) расширить и уточнить комплекс применяемых показателей, в том числе показателей, учитывающих атаки нулевого дня; (2) повысить эффективность методик выбора контрмер с точки зрения оперативности за счет применяемых алгоритмов, например, путем использования генетических алгоритмов, или путем распараллеливания задач; (3) добавить связь с базами контрмер; (4) учесть временной

аспект при выборе контрмер, а именно, время реализации контрмер и атак; (5) учесть исторический аспект при выборе контрмер, а именно, успешность реализации контрмеры в прошлом; (6) разработать систему визуализации предложенных показателей для удобства пользователя.

Выводы по главе 3

1) Разработана СОЗВК, которая позволяет снизить материальные потери от реализации компьютерных атак за счет формирования адекватной и актуальной оценки защищенности КС на основе доступных входных данных и выбора рациональных контрмер на основе сформированной оценки. Система отличается от существующих многоуровневым подходом и комплексным учетом различных характеристик объектов оценки, позволяющими формировать оценку защищенности в любой момент времени на основе всех доступных входных данных. Учет характеристик атакующего позволяет определить возможные шаги и цели атаки. Учет взаимосвязей между сервисами позволяет определить возможные потери от нарушения свойств безопасности ИТ активов и отследить распространение побочного ущерба в сети. Автоматизация процесса позволяет избежать ошибок оператора, сократить временные затраты и предоставляет оператору оценку защищенности в виде набора показателей. Учет событий безопасности, происходящих в КС, позволяет отслеживать изменение ситуации по защищенности во времени.

2) Представлена архитектура СОЗВК, реализующая разработанные методики в виде программного прототипа. В рамках прототипа реализованы все разработанные модели основных компонентов оценки защищенности, алгоритмы вычисления показателей защищенности, и методики оценки защищенности и выбора контрмер.

3) Для оценки эффективности применения методик оценки защищенности и выбора контрмер была проведена теоретическая и экспериментальная оценка эффективности. Критерием эффективности является выполнение требований по показателям таких свойств эффективности, как своевременность, обоснованность и ресурсопотребление. Оценка эффективности применения методики оценки защищенности КС и выбора контрмер в статическом и динамическом режимах работы показала, что предъявленные требования удовлетворены.

4) Предложены варианты по использованию и улучшению СОЗВК.

Заключение

Оценка защищенности и выбор защитных мер на основе адекватных количественных показателей в КС является важной и актуальной задачей ИБ. В диссертационной работе решена научная задача разработки комплекса моделей, методик и алгоритмов для оценки защищенности КС и выбора защитных мер для систем мониторинга безопасности и управления инцидентами, применение которых приведет к повышению эффективности процесса оценки защищенности и выбора контрмер. Основные результаты работы состоят в следующем:

1) Проанализирован процесс менеджмента риска ИБ и определено место оценки и обработки риска, выделены основные этапы данных процессов. Определено, что для КС организаций, для которых ИТ являются критичными, предпочтительной является детальная количественная оценка риска. Обоснована необходимость создания новых методик оценки и обработки риска на основе комплексного анализа данных из различных источников. Определены требования к разрабатываемым методикам.

2) Разработан комплекс показателей защищенности, включающий в себя отдельные показатели и связи между ними. Основным отличием предложенного комплекса является иерархический способ классификации показателей. Классификация осуществляется на основе входных данных, применяемых для вычисления показателей, этапов процесса анализа рисков и значений показателей. Система показателей защищенности интегрируется с SIEM-системами. Кроме того, она построена таким образом, что позволяет для каждой выделенной группы показателей получить оценку защищенности системы и выбрать защитные меры. И позволяет легко расширять список показателей для повышения точности и полноты анализа ситуации по защищенности.

3) Разработана методика оценки защищенности на основе графов атак и зависимостей сервисов и предложенного комплекса показателей. Основным отличием методики является иерархический характер, соответствующий уровням комплекса показателей, позволяющий в зависимости от имеющихся в наличии входных данных получить оценку текущей ситуации по защищенности, выраженную в форме адекватных количественных показателей, и уточнять оценку с появлением новых данных.

4) Разработаны алгоритмы вычисления показателей. В качестве входных данных для вычисления показателей применяются данные о сети и ее уязвимостях, существующие модели атак и зависимостей сервисов, разработанные модели

атакующих, событий и контрмер, экспертные оценки уязвимостей и контрмер, и оценки из открытых баз данных. Модели и методики используют такие стандарты протокола SCAP как CVE, CWE, CPE, CAPEC, CRE, ERI и CVSS. Это позволяет легко применять их в контексте систем мониторинга безопасности и управления инцидентами. Основным отличием от существующих алгоритмов является учет предложенного комплекса показателей, возможность как их отдельного применения в процессе оценки защищенности на каждом уровне иерархии комплекса показателей защищенности, так и их применения в рамках более высоких уровней для уточнения оценок предыдущих уровней (используя в качестве входных данных результаты предыдущих уровней).

5) Разработана методика выбора защитных мер, основанная на предложенном комплексе показателей. Методика включает сбор входных данных, вычисление показателей защищенности, выбор контрмер. Основными отличиями предложенной методики являются: применение предложенного комплекса показателей, выделение методик статического и динамического уровня (на первом уровне выбирается набор средств защиты, позволяющих реализовать контрмеры, на втором – конкретные контрмеры); совместное применение графов атак и зависимостей сервисов; и применимость для систем мониторинга безопасности и управления инцидентами.

6) Разработана архитектура и программная реализация системы оценки защищенности КС и выбора защитных мер. Основным отличием является применение оригинальных методик оценки защищенности КС и выбора защитных мер.

Полученные результаты соответствуют п. 9 Паспорта специальностей ВАК (технические науки) «Модели и методы оценки защищенности информации и информационной безопасности объекта» по специальности 05.13.19. Результаты являются развитием результатов работ [12–15, 17, 19–21, 23, 24, 27, 37, 43, 93–95, 97]. Они позволят снизить уровень возможных потерь организаций в результате компьютерных атак за счет постоянного отслеживания и пересчета показателей защищенности в соответствии с поступающими данными о событиях в системе и своевременного применения адекватных контрмер на основе доступных исходных данных. Практическая значимость результатов диссертационного исследования состоит в том, что они могут быть успешно реализованы в рамках компонента анализа защищенности и принятия решений активно распространяющихся систем мониторинга безопасности и управления инцидентами, что подтверждается их реализацией в ряде крупных российских и международных проектов. Апробация полученных результатов проводилась на 14 научно-технических конференциях. Основные результаты, полученные автором, опубликованы в 61 научных работах.

Перечень используемых сокращений и обозначений

ДМЗ – демилитаризованная зона
ИБ – информационная безопасность
ИС – информационная система
ИТ – информационные технологии
ИТТ –информационные и телекоммуникационные технологии
КС – компьютерные сети
ОС – операционная система
ПО – программное обеспечение
РФ – Российская Федерация
СМИБ – система менеджмента информационной безопасности
СОА – сервис-ориентированная архитектура
СОВ – системы обнаружения вторжений
СОЗВК – система оценки защищенности компьютерных сетей и выбора контрмер
СПВ – системы предотвращения вторжений
ALE – ожидаемые годовые потери (Annual Loss Expectancy)
CAPEC – «Общее перечисление и классификация шаблонов атак» (Common Attack Pattern Enumeration and Classification)
CCE – «Общее перечисление конфигураций» (Common Configuration Enumeration)
CCTA – центральное агентство по компьютерам и телекоммуникациям (Central Computer and Telecommunications Agency)
CPE – «Общее перечисление платформ» (Common Platform Enumeration)
CRE – «Общее перечисление защитных мер» (Common Remediation Enumeration)
CVE – «Общие уязвимости и дефекты» (Common Vulnerabilities and Exposures)
CVSS – «Общая система оценки уязвимостей» (Common Vulnerability Scoring System)
CWE – «Общее перечисление слабых мест» (Common Weakness Enumeration)
ERI – «Расширенная информация по защитным мерам» (Extended Remediation Information)
FIRST – Форум групп безопасности и реагирования на инциденты (Forum of Incident Response and Security Teams).
FISMA – федеральный акт по управлению информационной безопасностью (Federal Information Security Management Act)

FRAAP – методика качественного анализа рисков «Облегченный процесс анализа рисков» (Facilitated Risk Analysis and Assessment Process)

GTADM – модель атаки-защиты, основанная на теории игр (Game Theoretical Attack-Defense Model)

HRCM – иерархическая модель вычисления рисков (Hierarchical Risk Computing Model)

LDAP – «облегчённый протокол доступа к каталогам» (Lightweight Directory Access Protocol)

NIPC – Национальный центр защиты инфраструктуры США (National Infrastructure Protection Center)

NIST – национальный институт стандартизации и технологий США (U.S. National Institute of Information Standards and Technology)

NSA – Агентство национальной безопасности США (National Security Agency)

NVD – национальная база уязвимостей (National Vulnerability Database)

OCIL – Открытый язык отображения проверок безопасности» (Open Checklist Interactive Language)

OCTAVE – «Оперативная оценка критических угроз, активов и уязвимостей» (Operationally Critical Threat, Asset, and Vulnerability Evaluation)

OVAL – «Открытый язык спецификации уязвимостей и оценки» (Open Vulnerability and Assessment Language)

PCI DSS – стандарт безопасности данных индустрии платежных карт (Payment-Card Industry Data Security Standard)

RASQ – относительный коэффициент поверхности атаки (Relative Attack Surface Quotient)

ROI – оценка возврата инвестиций (Return on Investment)

SANS Institute's Critical Vulnerability Analysis Scale – шкала анализа критичных уязвимостей института SANS

SCAP – протокол автоматизации управления данными безопасности (Security Content Automation Protocol)

SIEM – системы мониторинга безопасности и управления инцидентами (Security Information and Events Management)

US-CERT – компьютерная группа реагирования на чрезвычайные ситуации (United states computer emergency readiness team)

XCCDF – «Расширяемый формат описания списков контроля конфигураций» (eXtensible Configuration Checklist Description Format)

XML – расширяемый язык разметки (Extensible Markup Language)

Список литературы и электронных ресурсов

1. **Агеев, А.** Positive SIEM [Электронный ресурс] / А. Агеев – Электрон. текстовые дан. – [Б. м. : б. и.]. – Режим доступа: http: // www.securitylab.ru / blog / personal / itsec / 139261.php (по состоянию на 13.05.2016).
2. Алгоритмы: построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М. : МЦНМО, 1999. – 960 с.
3. **Астахов, А.** Искусство управления информационными рисками [Текст] / А. М. Астахов. – М. : ДМК Пресс, 2010. – 312 с.
4. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий. Гостехкомиссия России. Руководящий документ [Электронный документ]. – Введ. 2002–06–19. – 56 с. – Режим доступа: http: // fstec.ru / component / attachments / download / 293 (по состоянию на 11.01.2017).
5. Выдержка из отчета Group-IB: Тенденции развития преступности в области высоких технологий 2014 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Group-IB, 2014. – Режим доступа: https: // new.landingi.com / uploads / ad335e01fdd23b3ff2d6 / landings / iUZba2WZSA3ab3axGe1c / assets / group-ib-annual-report-2014-short-rus.pdf (по состоянию на 11.01.2017).
6. **ГОСТ Р ИСО/МЭК 13335–1–2006.** Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий [Текст]. — Введ. 2006–12–19. — М. : Стандартинформ, 2007. — 19 с.
7. **ГОСТ Р ИСО/МЭК 13335-5-2006.** Информационная технология. Методы и средства обеспечения безопасности. Часть 5. Руководство по менеджменту безопасности сети [Текст]. — Введ. 2007–06–01. — М. : Стандартинформ, 2007. — 22 с.
8. **ГОСТ Р ИСО/МЭК 27001-2006.** Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования [Текст]. — Введ. 2006–12–27. — М.: Стандартинформ, 2008. — 26 с.
9. **ГОСТ Р ИСО/МЭК 27004-2011.** Информационная технология. Методы и средства обеспечения безопасности. Менеджмент информационной безопасности. Измерения [Текст]. — Введ. 2011–12–01. — М. : Стандартинформ, 2012. — 56 с.

10. **ГОСТ Р ИСО/МЭК 27005-2010.** Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности [Текст]. — Введ. 2010–11–30. — М. : Стандартинформ, 2011. – 47 с.
11. **ГОСТ Р ИСО/МЭК 31010-2011.** Менеджмент риска. Методы оценки риска [Текст]. — Введ. 2011–12–01. — М. : Стандартинформ, 2012. — 74 с.
12. **Дойникова, Е. В.** Динамическое оценивание защищенности компьютерных сетей в SIEM-системах [Текст] / Е. В. Дойникова, И. В. Котенко, А. А. Чечулин // Безопасность информационных технологий. – М.: ВНИИПВТИ, 2015. – № 3. – С. 33–42.
13. **Дойникова, Е. В.** Методики и программный компонент оценки рисков на основе графов атак для систем управления информацией и событиями безопасности [Текст] / Е. В. Дойникова, И. В. Котенко // Информационно-управляющие системы. – СПб. : Политехника, 2016. – № 5. – С. 54–65. – doi:10.15217/issn1684-8853.2016.5.54.
14. **Дойникова, Е. В.** Отслеживание текущей ситуации и поддержка принятия решений по безопасности компьютерной сети на основе системы показателей защищенности [Текст] / Е. В. Дойникова, И. В. Котенко // Изв. вузов. Приборостроение. – СПб. : СПбГУ ИТМО, 2014. – Т. 57, № 10. – С. 72–77. – ISSN 0021-3454.
15. **Дойникова, Е. В.** Показатели и методики оценки защищенности компьютерных сетей на основе графов атак и графов зависимостей сервисов [Текст] / Е. В. Дойникова // Труды СПИИРАН. – СПб. : Наука, 2013. – Вып. 3 (26). – С. 54–68.
16. **Калашников, А.О.** Информационные риски флуд-атакуемых компьютерных систем [Текст] / А.О. Калашников, А.Г. Остапенко, Г.А. Остапенко, М.В. Бурса, В.В. Бутузов –. Воронеж : ООО «Издательство Научная книга», 2015. – 160 с.
17. **Котенко, И. В.** Анализ защищенности автоматизированных систем с учетом социо-инженерных атак [Текст] / И. В. Котенко, М. В. Степашкин, Е. В. Дойникова // Проблемы информационной безопасности. Компьютерные системы. – СПб. : СПбПУ, 2011. – № 3. – С. 40–57.
18. **Котенко, И. В.** Анализ протокола автоматизации управления данными безопасности SCAP [Текст] / И. В. Котенко, Е. В. Дойникова // Защита информации. Инсайд. – СПб, 2012. – № 2. – С. 56–63.
19. **Котенко, И. В.** Вычисление и анализ показателей защищенности на основе графов атак и зависимостей сервисов [Текст] / И. В. Котенко, Е. В. Дойникова //

Проблемы информационной безопасности. Компьютерные системы. – СПб. : СПбПУ, 2014. – № 2. – С. 19–36.

20. **Котенко, И. В.** Динамический перерасчет показателей защищенности на примере определения потенциала атаки [Текст] / И. В. Котенко, Е. В. Дойникова, А. А. Чечулин // Труды СПИИРАН. – СПб.: Наука, 2013. – Вып. 7 (30). – С. 26–39. – ISSN: 2078-9181.
21. **Котенко, И. В.** Методика выбора контрмер в системах управления информацией и событиями безопасности [Текст] / И. В. Котенко, Е. В. Дойникова // Информационно-управляющие системы. – СПб. : Политехника, 2015. – № 3. – С. 60–69. – doi:10.15217/issn1684-8853.2015.3.60.
22. **Котенко, И. В.** Методы оценивания уязвимостей: использование для анализа защищенности компьютерных систем [Текст] / И. В. Котенко, Е. В. Дойникова // Защита информации. Инсайд. – СПб., 2011 – № 4. – С. 74–81.
23. **Котенко, И. В.** Общее перечисление и классификация шаблонов атак (CAPEC): описание и примеры применения [Текст] / И. В. Котенко, Е. В. Дойникова, А. А. Чечулин // Защита информации. Инсайд. – СПб., 2012. – № 4. – С. 54–66.
24. **Котенко, И. В.** Оценка защищенности информационных систем на основе построения деревьев социо-инженерных атак [Текст] / И. В. Котенко, М. В. Степашкин, Д. И. Котенко, Е. В. Дойникова // Изв. вузов. Приборостроение. – СПб. : СПбГУ ИТМО, 2011. – Т. 54, № 12. – С. 5–9. – ISSN 0021-3454.
25. **Котенко, И. В.** Оценка рисков в компьютерных сетях критических инфраструктур [Текст] / И. В. Котенко, И.Б. Саенко, Е. В. Дойникова // Инновации в науке: материалы XVI международной заочной научно-практической конференции (Новосибирск, 2013). – Новосибирск: СибАК, 2013. – № 16-1. – С. 84–88.
26. **Котенко, И. В.** Система оценки уязвимостей CVSS и ее использование для анализа защищенности компьютерных систем [Текст] / И. В. Котенко, Е. В. Дойникова // Защита информации. Инсайд. – СПб., 2011 – № 5. – С. 54–60.
27. **Котенко, И. В.** Метрики безопасности для оценки уровня защищенности компьютерных сетей [Текст] / И. В. Котенко, М. В. Степашкин // Защита информации. Инсайд. – СПб., 2006. – № 3.
28. Меры защиты информации в государственных информационных системах. Методический документ [Электронный документ]. – Утвержден ФСТЭК России 2014–02–11. – 176 с. – Режим доступа: http: // fstec.ru / component / attachments / download / 675 (по состоянию на 11.01.2017).

29. **Новикова, Е. С.** Проектирование компонента визуализации для автоматизированной системы управления информационной безопасностью [Текст] / Е. С. Новикова, И. В. Котенко // Информационные технологии. – Новые технологии, 2013. – № 9. – С. 32–36.
30. **Орлик С.** Введение в программную инженерию и управление жизненным циклом ПО. Программная инженерия. Программные требования [Текст] / С. Орлик, Ю. Булуй. – 2004–2005.
31. Основы теории управления в системах военного назначения. Часть II. Учебное пособие [Текст] / Е. А. Карпов, И. В. Котенко, А. В. Боговик, И. С. Ковалев, А. Н. Забело, С. С. Загорулько, В. В. Олейник ; под редакцией А. Ю. Рунеева и И. В. Котенко. – СПб.: ВУС, 2000. – 200 с.
32. Отчет McAfee об угрозах за первый квартал 2013 года [Электронный ресурс] / McAfee Labs. – Электрон. текстовые дан. – [Б. м. : б. и.] – Режим доступа: http: // www.mcafee.com / ru / resources / reports / rp-quarterly-threat-q1-2013.pdf (по состоянию на 19.12.2016).
33. Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. – [Б. м. : б. и.]. – Режим доступа: http: // vak.ed.gov.ru / documents / 10179 / 0 / %D0%9F%D0%B5%D1%80%D0%B5%D1%87%D0%B5%D0%BD%D1%8C%20%D0%92%D0%90%D0%9A%2003.06.2016.pdf / dbb423d3-5bfe-4197-bd5d-d8825c07f2a9 (по состоянию на 16.09.2016).
34. Продукт MaxPatrol SIEM [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. – [Б. м. : б. и.]. – Режим доступа: http: // www.ptsecurity.ru / products / mpsiem (по состоянию на 14.05.2016).
35. Сервис-ориентированная архитектура (SOA) и архитектура, управляемая моделями (MDA) [Электронный ресурс] : Студопедия. – Электрон. текстовые данные. – [Б. м. : б. и.], 2014. – Режим доступа: http: // studopedia.info / 1-118261.html (по состоянию на 20.10.2015)
36. Система сетевого планирования и управления. Методические указания к проведению практических занятий для студентов экономических и технических специальностей всех форм обучения [Электронный документ] / Составитель Г. М. Охезина. – Электрон. текстовые данные и граф. данные. – [Б. м. : б. и.]. – Режим

доступа: http: // www.nntu.ru/ RUS / fakyl/ VECH / metod / orgprod1 / part5.htm (по состоянию на 30.08.2016)

37. **Степашкин, М. В.** Модели и методика анализа защищенности компьютерных сетей на основе построения деревьев атак [Текст]: диссертация кандидата технических наук : 05.13.11, 05.13.19 / Степашкин, Михаил Викторович; [Место защиты: С.-Петерб.]. – Санкт-Петербург, 2007. – 196 с. : ил. Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.
38. Стратегия развития отрасли информационных технологий в Российской Федерации на 2014 – 2020 годы и на перспективу до 2025 года [Электронный документ] / Электрон. текстовые данные и граф. данные. – [Б. м. : б. и.]. – Режим доступа: http: // government.ru / media / files / 41d49f3cb61f7b636df2.pdf (по состоянию на 09.01.2017).
39. Таблица значений функции Лапласа. Интерактивный справочник [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.webmath.ru / poleznoe / table_laplasa.php (по состоянию на 30.08.2016).
40. Управление информационными рисками. Экономически оправданная безопасность [Текст] / С.А. Петренко, С.В. Симонов. – М.: АйТи Пресс, 2004. – 384 с.
41. **Федотова, А. А.** Основы сетевого планирования и управления в физической культуре и спорте [Текст]. Учебно-методическое пособие / А. А. Федотова, Ю. Н. Федотов, В. А. Платонова. – СПб: СПбГУ ИТМО, 2006. – 15 с.
42. **Черешкин, Д.С.** Управление рисками и безопасностью [Текст] / Д.С. Черешкин: Труды Института системного анализа РАН. – URSS, 2009. – 288 с. – ISBN 978-5-9710-0255-0.
43. **Чечулин, А. А.** Построение и анализ деревьев атак на компьютерные сети с учетом требования оперативности [Текст] : диссертация кандидата технических наук : 05.13.19 / Чечулин Андрей Алексеевич; [Место защиты: С.-Петерб. ин-т информатики и автоматизации РАН]. – Санкт-Петербург, 2013. – 152 с. : ил. Методы и системы защиты информации, информационная безопасность. Хранение: 61 14-5/933.
44. **Шелестова, О**. Обзор. Конкурс «Чего нам не хватает в SIEM»: победители и разбор полетов [Электронный ресурс] / О. Шелестова ; Новости сайта SecurityLab.ru. – 26 июня, 2014 (по состоянию на 14.06.2016).

45. A Guide for Government Agencies Calculating Return on Security Investment. Version 2.0 [Текст] / Government Chief Information Office (GCIO). – Lockstep Consulting, 2014. – 33 p.
46. **Ahmed, M. S.** A novel quantitative approach for measuring network security [Текст] / M. S. Ahmed, E. Al-Shaer, L. Khan. – Proceedings of the INFOCOM'08. – [Б. м. : б. и.], 2008. – P.1957–1965.
47. **Artz, M.** NetSPA, a network security planning architecture [Текст] : Master's thesis / M. Artz. – Massachusetts Institute of Technology, 2002.
48. Atos Societas Europaea [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – [Б. м. : б. и.]. – Режим доступа: http: // atos.net / en-us / home.html (по состоянию на 11.11.2015)
49. **Axelrod, C. W.** Accounting for value and uncertainty in security metrics [Текст] / C. W Axelrod // Information Systems Control Journal. – [Б. м. : б. и.], 2008. – vol.6. – P.1–6.
50. **Balepin, I.** Using specification-based intrusion detection for automated response [Текст] / I. Balepin, S. Maltsev, J. Rowe, K. Levitt // Proceedings of the sixth International Symposium on Recent Advances in Intrusion Detection (RAID). – [Б. м. : б. и.], 2003. – P. 136–154.
51. **Barnum, S.** Common Attack Pattern Enumeration and Classification (CAPEC) [Текст] / Schema Description. – [Б. м. : б. и.], 2008.
52. **Bayne, J.** An Overview of Threat and Risk Assessment [Электронный ресурс] / SANS Institute InfoSec Reading Room. – Электрон. текстовые данные и граф. данные. – [Б. м. : б. и.], 2002. – P. 9. – Режим доступа: https: // www.sans.org / reading-room / whitepapers / auditing / overview-threat-risk-assessment-76 (по состоянию на 19.10.2015)
53. **Bursztein, E.** Using strategy objectives for network security analysis [Текст] / E. Bursztein, J. C. Mitchell // Information Security and Cryptology ; series: Lecture Notes in Computer Science. – Volume 6151. – Springer Berlin Heidelberg, 2010. – P. 337–349.
54. C&A Systems Security Ltd : company web-site [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.riskworld.net (по состоянию на 19.12.2016).
55. Information Security Metrics. State of the Art [Текст] / DSV Report; writer: R. Barabanov, eds.: S. Kowalski, L. Yngström. – No 11-007. – [Б. м. : б. и.], 2011.

56. **Caralli, R. A.** Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process [Текст]. Technical Report. / R. A. Caralli, J. F. Stevens, L. R. Young, W. R. Wilson. – Software Engineering Institute, 2007. – P. 154.
57. **Chunlu, W**. A novel comprehensive network security assessment approach [Текст] / W. Chunlu, W. Yancheng, D. Yingfei Z. Tianle // Proceedings of the 2011 IEEE International Conference on Communications (Kyoto). – IEEE, 2011. – P. 1–6.
58. Cisco Secure Intrusion Detection System [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.cisco.com / en / US / products / hw / vpndevc / ps4077 / prod_eol_notice09186a008009230e.html. (по состоянию на 8.11.2015)
59. Common Attack Pattern Enumeration and Classification (CAPEC) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: https: // capec . mitre . org (по состоянию на 19.12.2016).
60. Common Platform Enumeration (CPE) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // cpe.mitre.org (по состоянию на 19.12.2016).
61. Common Vulnerabilities and Exposures (CVE) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // cve.mitre.org (по состоянию на 19.12.2016).
62. Common Weakness Enumeration (CWE) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: https: // cwe.mitre.org / data / index.html (по состоянию на 19.10.2015).
63. Comodo [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.comodo.com. (по состоянию на 8.11.2015).
64. CRAMM [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.cramm.com (по состоянию на 19.12.2016).
65. **Cremonini, M.** Evaluating information security investments from attackers perspective: the Return-On-Attack (ROA) [Текст] / M. Cremonini, P. Martini // Proceedings of the Fourth Workshop on the Economics of Information Security (2-3 June 2005). – [Б. м. : б. и.], 2005.
66. **Dacier, M.** Quantitative Assessment of Operational Security-Models and Tools [Текст]. LAAS Research Report 96493 / M. Dacier, Y. Deswarte et al. – 1996.
67. **Dantu, R.** Network risk management using attacker profiling [Текст] / R. Dantu, P. Kolan, J. Cangussu // Security and Communication Networks. – [Б. м. : б. и.], 2009. – Vol. 2, No.1. – P. 83–96.

68. Endian [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.endian.com (по состоянию на 08.11.2015).
69. FreeNATS [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.purplepixie.org / freenats (по состоянию на 8.11.2015).
70. **Frigault, M.** Measuring network security using dynamic Bayesian network / M. Frigault, L. Wang, A. Singhal, S. Jajodia // Proceedings of the ACM Workshop on Quality of Protection (October 2008). – [Б. м. : б. и.], 2008.
71. **Gerard, T. M.** Common Remediation Enumeration (CRE) Version 1.0 (Draft). NIST Interagency Report 7831 (Draft) / T. M. Gerard, D. Waltermire, J. O. Baker ; National Institute of Standards and Technology, U.S. Department of Commerce. – [Б. м. : б. и.], 2011.
72. Global Threat Intelligence Report / NTTGroup; NTTInnovationInstitute 1 LLC. – 2014. – 67 p.
73. Grance, T. [Presentation]. Automating Compliance with Security Content Automation Protocol // T. Grance ; NIST. – 2008.
74. **Granadillo, G. G.** Individual countermeasure selection based on the return on response investment index. / G. G. Granadillo, H. Debar, G. Jacob, M. Achemlal // LNCS, Computer Network Security, proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012 (St. Petersburg, Russia, October 17-19, 2012). – Vol.7531. – Springer, 2012. – P.156–170. – DOI: 10.1007/978-3-642-33704-8_14.
75. **He, W.** A network security risk assessment framework based on game theory / W. He, C. Xia, C. Zhang, Y. Ji, X. Ma // Proceedings of the Second International Conference on Future Generation Communication and Networking, FGCN '08 (Hainan Island, 13-15 Dec. 2008). – Volume 2. – IEEE, 2009. – P. 249–253.
76. **Henning, R.** Security Metrics / R. Henning, et al. ; MITRE // Proceedings of the Workshop on Information Security System, Scoring and Ranking (Williamsburg, Virginia). – 2002.
77. **Hoo, K. J. S.** How Much is Enough? A Risk-Management Approach to Computer Security [Текст] : PhD thesis / K. J. S. Hoo. – Stanford University, 2000.
78. **Howard, J.** A Common Language for Computer Security Incidents. SANDIA Report / J. Howard, T. Longstaff. – SAND98-8667. – 1998.
79. **Howard, M.** Measuring relative attack surfaces [Электронный ресурс] / M. Howard, J. Pincus, J. M. Wing // Proceedings of Workshop on Advanced Developments in Software and Systems Security (Taipei. 2003). – Режим доступа: http: //

www.cs.cmu.edu / ~wing/publications/Howard-Wing03.pdf (по состоянию на 19.12.2016).

80. **Idika, N. C.** Characterizing and aggregating attack graph-based security metric : PhD Thesis / N. C. Idika. – Purdue University, 2010. – 131 p.
81. **Ingols, K.** Practical attack graph generation for network defense [Текст] / K. Ingols, R. Lippmann, K. Piwowarski // Proceedings of 22nd Annual Conference on the Computer Security Applications (Miami Beach, FL, 2006). – IEEE, 2006. – P. 121 – 130.
82. Internet Security Threat Report 2013 [Текст] / Symantec Corporation. – Volume 18. 2013. – 58 p.
83. Intrust [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.quest.com / intrust (по состоянию на 8.11.2015)
84. **ISO/IEC 17799:2005**. Information technology. – Security techniques. – Code of practice for information security management. – Введ. 2005-06-15.
85. **Jahnke, M.** Graph-based metrics for intrusion response measures in computer networks / M. Jahnke, C. Thul, P. Martini // Proceedings of the IEEE Workshop on Network Security (2007).
86. **Johnson, C.** Enterprise Remediation Automation [Текст] / C. Johnson ; NIST // Proceedings of the IT Security Automation Conference (September 27-29, 2010).
87. **Kanoun, W.** Automated reaction based on risk analysis and attackers skills in intrusion detection systems [Текст] / W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, J. Araujo // Proceedings of the Third International Conference on Risks and Security of Internet and Systems (28-30 Oct. 2008). – P. 117–124.
88. Kaspersky [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // usa.kaspersky.com / products-services / home-computer-security / anti-virus (по состоянию на 08.11.2015).
89. **Kheir, N.** A service dependency model for cost-sensitive intrusion response / N. Kheir, N. Cuppens-Boulahia, F. Cuppens, H. Debar // Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS'10). – Vol. 6345. – 2010. – P. 626–642.
90. **Kheir, N.** Cost evaluation for intrusion response using dependency graphs [Текст] / N. Kheir, H. Debar, N. Cuppens-Boulahia, F. Cuppens, J. Viinikka / Proceedings of the International Conference on Network and Service Security (Paris, 24-26 June 2009). – IEEE, 2009. – P. 1–6.
91. **Kheir, N.** Response policies & counter-measures: Management of service dependencies and intrusion and reaction impacts : PhD Thesis / N. Kheir. – Telecom Bretagne, 2010.

92. **Kotenko, I.** Attack graph based evaluation of network security / I. Kotenko, M. Stepashkin // Proceedings of the 10th IFIP Conference on Communications and Multimedia Security (Heraklion, Greece, 2006). – 2006. – P. 216–227.
93. **Kotenko, I.** Countermeasure selection in SIEM systems based on the integrated complex of security metrics / I. Kotenko, E. Doynikova // Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing. – 2015. – P. 567–574.
94. **Kotenko, I.** Evaluation of computer network security based on attack graphs and security event processing / I. Kotenko, E. Doynikova // Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. – 2014. – Vol.5, No.3. – P. 14–29.
95. **Kotenko, I.** Security evaluation models for cyber situational awareness / I. Kotenko, E. Doynikova // Proceedings of the 2014 IEEE 6th International Symposium on Cyberspace Safety and Security (Paris, France, 2014). – 2014. – Los Alamitos, California: IEEE Computer Society, 2014. – P. 1229–1236.
96. **Kotenko, I.** Security risks management in the internet of things based on fuzzy logic inference / I. Kotenko, I. Saenko, S. Ageev // Proceedings of the 2015 IEEE International Symposium on Recent Advances of Trust, Security and Privacy in Computing and Communications in conjunction with the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Helsinki, Finland, 2015). – 2015. – Los Alamitos, California: IEEE Computer Society, 2015. – P. 654–659.
97. **Kotenko, I.** The CAPEC based generator of attack scenarios for network security evaluation / I. Kotenko, E. Doynikova // Proceedings of the IEEE 8th International Conference on “Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications” (Warsaw, Poland, 2015). – 2015. – P. 436–441.
98. **Lippmann, R. P.** Validating and restoring defense in depth using attack graphs [Текст] / R. P. Lippmann et al // Proceedings of MILCOM 2006 (Washington, DC).
99. **Liu, Y.** Network vulnerability assessment using Bayesian networks / Y. Liu, H. Man // Proceedings of the SPIE. – vol. 5812. – 2005. – P. 61-71.
100. **Lorenzo, J. M.** Alienvault users manual. Version 1.0 / J. M. Lorenzo ; Alienvault LC. – 2011. – P. 225.
101. **Man, D.** A quantitative evaluation model for network security [Текст] / D. Man, W. Yang, Y. Yang, W. Wang, L. Zhang // Proceedings of the 2007 International

Conference on Computational Intelligence and Security (15-19 Dec. 2007). – P. 773–777.

102. **Manadhata, P. K.** A formal model for a system's attack surface [Электронный ресурс] / P. K. Manadhata, D. K. Kaynar, J. M. Wing. – Pittsburgh, PA: Carnegie Mellon University, 2007. – Режим доступа: http: // www.cs.cmu.edu / ~wing / publications / CMU-CS-07-144.pdf (по состоянию на 19.12.2016).

103. **Manadhata, P. K.** An attack surface metric / P. K. Manadhata, J. M. Wing // IEEE Transactions on Software Engineering (June 2010). – 2010.

104. **Manadhata, P. K.** Measuring a system’s attack surface [Электронный ресурс] / P. K. Manadhata. – PA: Carnegie Mellon University, 2004. – Режим доступа: http: // reports-archive.adm.cs.cmu.edu / anon / 2004 / CMU-CS-04-102.pdf (по состоянию на 19.12.2016).

105. MASSIF FP7 Project. MAnagement of Security information and events in Service Infrastructures [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – EC FP7-257475. – Режим доступа: http: // www.massif-project.eu (по состоянию на 29.10.2015).

106. **Mayer, A.** Operational security risk metrics: definitions, calculations, visualizations [Presentation] / A. Mayer ; RedSeal Systems, Inc. // Metricon 2.0. – 2007.

107. **McGuire G. T.** Common Remediation Enumeration (CRE) Version 1.0 (Draft). NIST Interagency Report 7831 (Draft) / G. T. McGuire, D. Waltermire, J. O. Baker ; NIST. – 2011.

108. **McIntyre, A.** I3P Research report No. 12. Security metrics tools final report [Электронный документ] / A. McIntyre, B. Becker, D. Bodeau, B. Gennert, C. Glantz, L. R. O’Neil, J. R. Santos, M. Stoddard // Institute for Information Infrastructure Protection. – [Б. м. : б. и.], 2007. – Режим доступа: http: // citeseerx.ist.psu.edu / viewdoc / download?doi=10.1.1.170.1610&rep=rep1&type=pdf (по состоянию на 24.03.2016).

109. **Mell, P.** A Complete Guide to the Common Vulnerability Scoring System (CVSS) Version 2.0 [Электронный документ] / P. Mell, K. Scarforne, S. Romanosky. – [Б. м. : б. и.], 2007. – Режим доступа: http: // www.first.org / cvss / cvss-guide.html (по состоянию на 19.12.2016).

110. Microsoft Security Response Center Security Bulletin Severity Rating System / Microsoft Corporation. – 2002.

111. MITRE Website [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.mitre.org (по состоянию на 19.12.2016).

112. MITRE. CWE Schema Documentation [Электронный документ] / Электрон. текстовые данные и граф. данные. –17 р. – Режим доступа: https: // cwe.mitre.org / documents / schema / schema_d9.pdf (по состоянию на 19.10.2015).

113. National Infrastructure Protection Center. CyberNotes [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Issue 4-99. – [Б. м. : б. и.], 1999. – Режим доступа: http: // www.irational.org / APD / IPC / cyberissue4.pdf (по состоянию на 19.10.2015).

114. nCircle Vulnerability Scoring System [Электронный документ] / nCircle Network Security Inc. – 2009. – 12 р. – Режим доступа: http: // www.usdatavault.com / library / ncircle_vulnerability_scoring.pdf (по состоянию на 19.12.2016).

115. Nessus vulnerability scanner [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.tenable.com / products / nessus-vulnerability-scanner (по состоянию на 29.10.2015).

116. NetCrunch [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.adremsoft.com / netcrunch (по состоянию на 8.11.2015).

117. NMap reference guide [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // nmap.org / book / man.html (по состоянию на 8.11.2015).

118. **Noel, S.** Efficient minimum-cost network hardening via exploit dependency graphs / S. Noel, S. Jajodia, B. O'Berry, M. Jacobs // Proceedings of the 19th Annual Computer Security Applications Conference (8-12 Dec. 2003). – IEEE, 2003. – P. 86–95.

119. NVD website [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: https: // nvd.nist.gov (по состоянию на 19.12.2016).

120. OCTAVE [Электронный ресурс] / CERT website. – Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.cert.org / resilience / products-services / octave / index.cfm (по состоянию на 19.10.2015).

121. **Olsson, T.** Assessing security risk to a network using a statistical model of attacker community competence / T. Olsson // Proceedings of the 11th international conference on Information and Communications Security. – 2009. – P. 308–324.

122. **Ou, X.** MULVAL: A logic based network security analyzer / X. Ou, S. Govindavajhala, A. W. Apple // Proceedings of the 14th USENIX Security Symposium (CA, USA). – Volume 14. – 2005. 8 p.

123. PCI DSS (PCI Data Security Standard). Version 3.2 / 2006-2016 PCI Security Standards Council. – Введ. 04.2010. – 137 p.

124. **Peltier, T. R.** Information security risk analysis, Third Edition. / T. R. Peltier. – CRC Press, 2010. 456 p.

125. **Poolsappasit, N.** Dynamic security risk management using Bayesian attack graphs / N. Poolsappasit, R. Dewri, I. Ray // IEEE Transactions on Dependable and Security Computing. – 2012. – Vol.9, No.1 – P. 61–74.

126. RiskWatch [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.riskwatch.com (по состоянию на 12.12.2016).

127. Security and Privacy Controls for Federal Information Systems and Organizations. NIST Special Publication 800-53. Revision 4 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – NIST, 2013. Режим доступа: http: // nvlpubs.nist.gov / nistpubs / SpecialPublications / NIST.SP.800-53r4.pdf (по состоянию на 12.12.2016).

128. **Seddigh, N.** Current trends and advances in information assurance metrics / N. Seddigh, P. Pieda, A. Matrawy, B. Nandy, I. Lambadaris, A. Hatfield // Proceedings of the 2nd Annual Conference on Privacy, Security and Trust (Fredericton, NB, Oct. 2004). – 2004.

129. **Singhal, A.** Security risk analysis of enterprise networks using probabilistic attack graphs. NIST Interagency Report 7788. / A. Singhal, X. Ou. – Gaithersburg: National Institute of Standards and Technology, 2011. 24 p.

130. Snort [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: https: // www.snort.org (по состоянию на 8.11.2015).

131. **Stakhanova, N.** A cost-sensitive model for preemptive intrusion response systems / N. Stakhanova, S. Basu, J. Wong. // Proceedings of the 21st International Conference on Advanced Networking and Applications. – 2007.

132. **Strasburg, C.** Intrusion response cost assessment methodology / C. Strasburg, N. Stakhanova, S. Basu, J. S. Wong // Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (NY, USA, 2009). – 2009. – P. 388–391.

133. **Swanson, M.** Security metrics guide for information technology systems. NIST Special Publication 800-55 / M. Swanson, N. Bartol, J. Sabato, J. Hash, L. Graffo. – 2003.

134. The Center for Internet Security. The CIS Security Metrics [Текст]. – The Center for Internet Security, 2009. – 175 p.

135. Threat Intelligence Report 2012-2013 H1 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Group-IB, 2013. – Режим доступа: http: // report2013.group-ib.ru (по состоянию на 19.12.2016).

136. **Toth, T.** Evaluating the impact of automated intrusion response mechanisms / T. Toth, C. Kruegel // Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC). – 2002.– P. 301–310.

137. US-CERT [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http: // www.us-cert.gov (по состоянию на 19.12.2016).

138. Using Attack Surface Area And Relative Attack Surface Quotient To Identify Attackability. Customer Information Paper [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Ernst & Young LLP, 2003. – 8 p. Режим доступа: https: // www.microsoft.com / windowsserver2003 / docs / AdvSec.pdf (по состоянию на 19.10.2015).

139. **Vaughn, R.** Information assurance measures and metrics: State of Practice and Proposed Taxonomy / R. Vaughn, R. Henning, A. Siraj // Proceedings of the 36th Hawaii Int. Conf. on System Sciences (HICSS 03). – 2003.

140. **Visintine, V**. Global information assurance certification paper [Электронный ресурс]. SANS Institute 2003 / V. Visintine. –13 p. – Режим доступа: http: // www.giac.org / paper / gsec / 3156 / introduction-information-risk-assessment / 105258 (по состоянию на 19.10.2015).

141. **Wang, L.** An Attack Graph-Based Probabilistic Security Metric / L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia // Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security. – Heidelberg: Springer-Verlag Berlin, 2008. – P. 283–296. – DOI: 10.1007/978-3-540-70567-3_22.

142. Web Services Glossary. W3C Working Group Note 11 February 2004 [Электронный ресурс] / eds.: H. Haas, A. Brown. – Режим доступа: http: // www.w3.org / TR / ws-gloss (по состоянию на 20.10.2015).

143. **Williams, L.** GARNET: A Graphical Attack Graph and Reachability Network Evaluation Tool [Текст] / L. Williams. – Massachusetts Institute of Technology, 2008.

144. Wireshark vulnerability scanner [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: https: // www.wireshark.org (по состоянию на 29.10.2015).

145. **Wu, Y.-S.** Automated adaptive intrusion containment in systems of interacting services / Y.-S. Wu, B. Foo, Y.-C. Mao, S. Bagchi, E. H. Spafford // Computer Networks: The International Journal of Computer and Telecommunications Networking. – 2007. – Vol.51. – P. 1334–1360.

146. X-Force [Электронный ресурс] / Электрон. текстовые данные и граф. данные. – Режим доступа: http:// xforce.iss.net (по состоянию на 02.04.2015).

Приложение А – Схема алгоритма определения локальных вероятностей

Приложение Б – Блок-схема алгоритма определения условных вероятностей

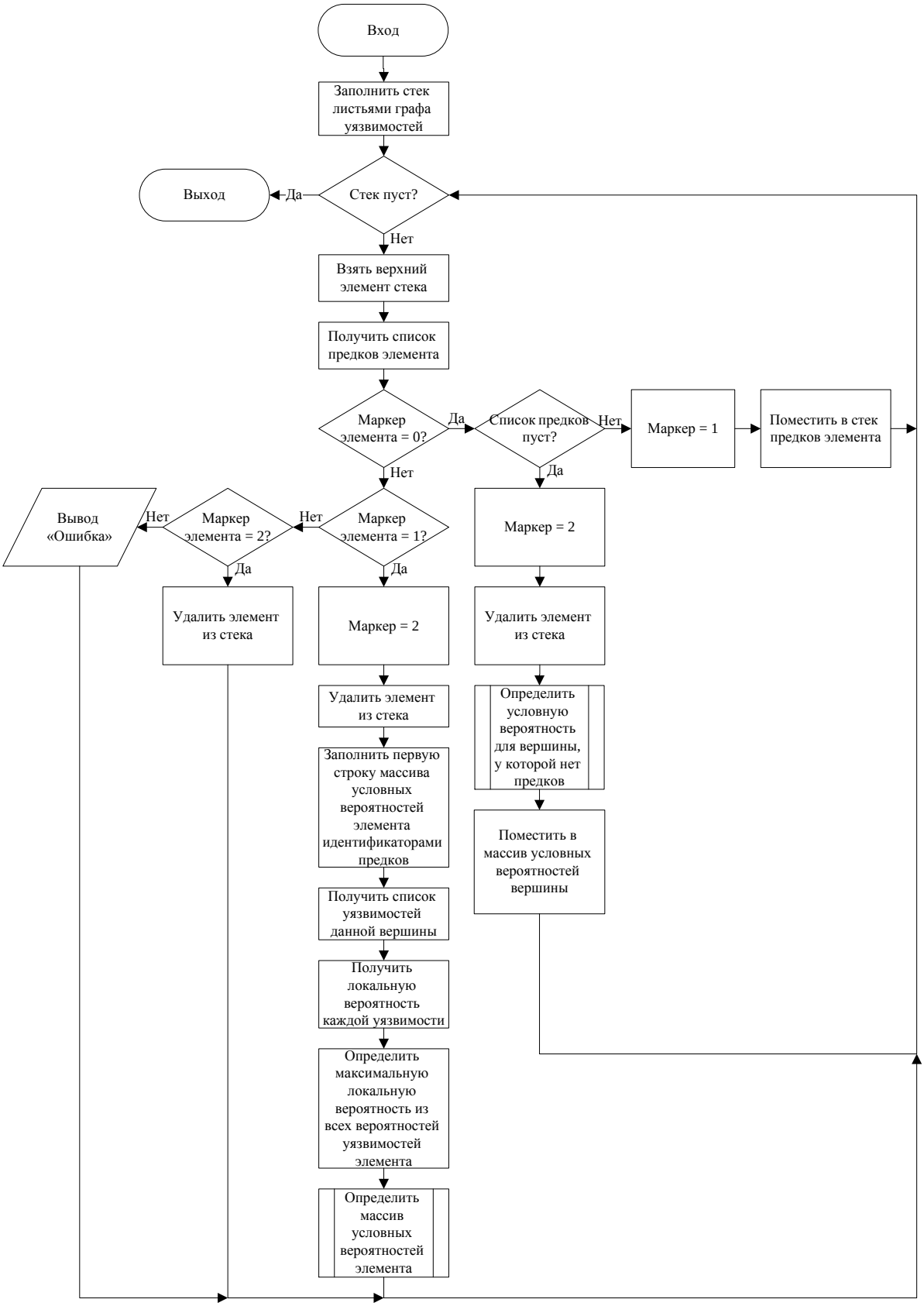
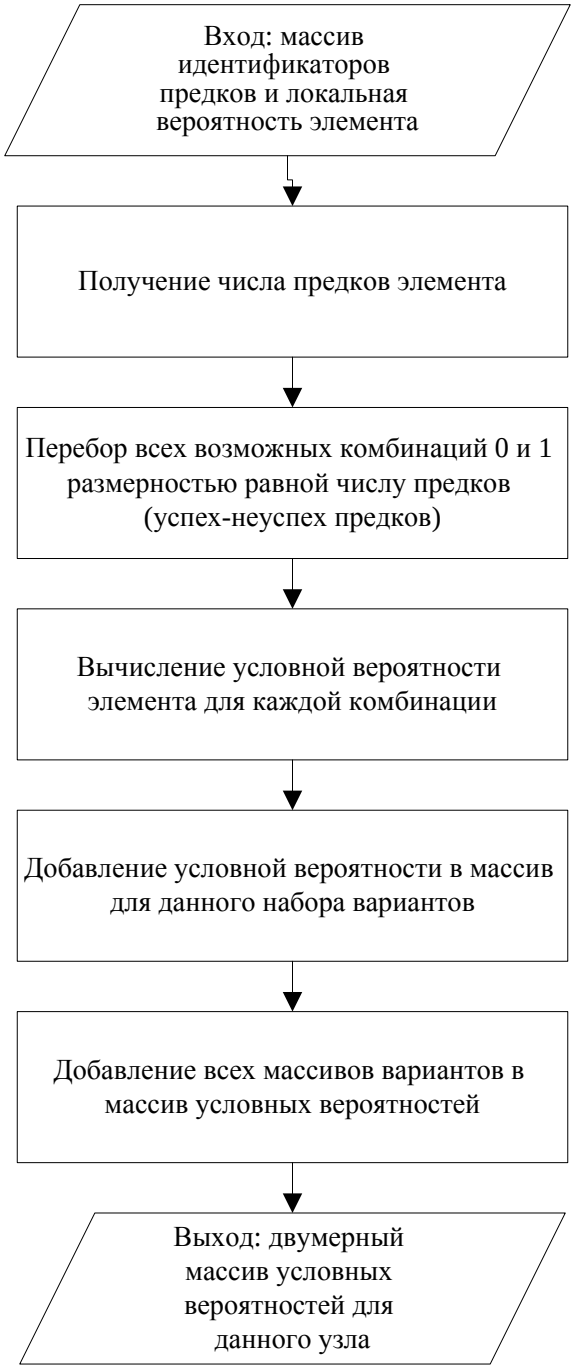


Схема функции определения массива условных вероятностей вершины



Комментарий: первая строка двумерного массива условных вероятностей узла представляет собой массив идентификаторов предков узла, последний элемент строки – идентификатор узла. Остальные строки представляют собой комбинации 0 и 1, отображающие все возможные варианты состояний для предков узла, последний элемент строк – условная вероятность узла для соответствующей комбинации состояний его предков.

Схема функции определения условной вероятности вершины у которой нет предков

Приложение В – Схема алгоритма определения безусловных вероятностей

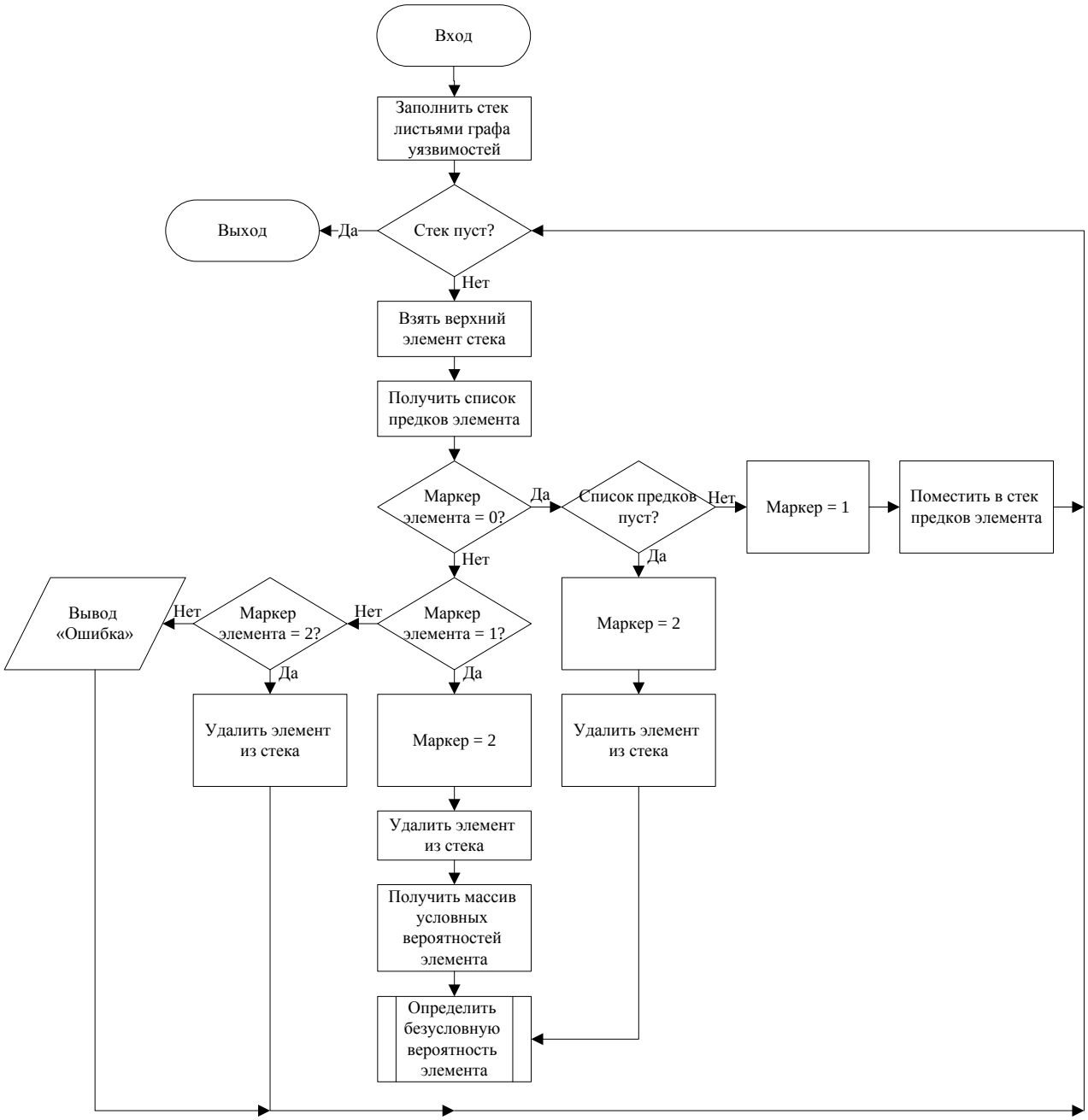
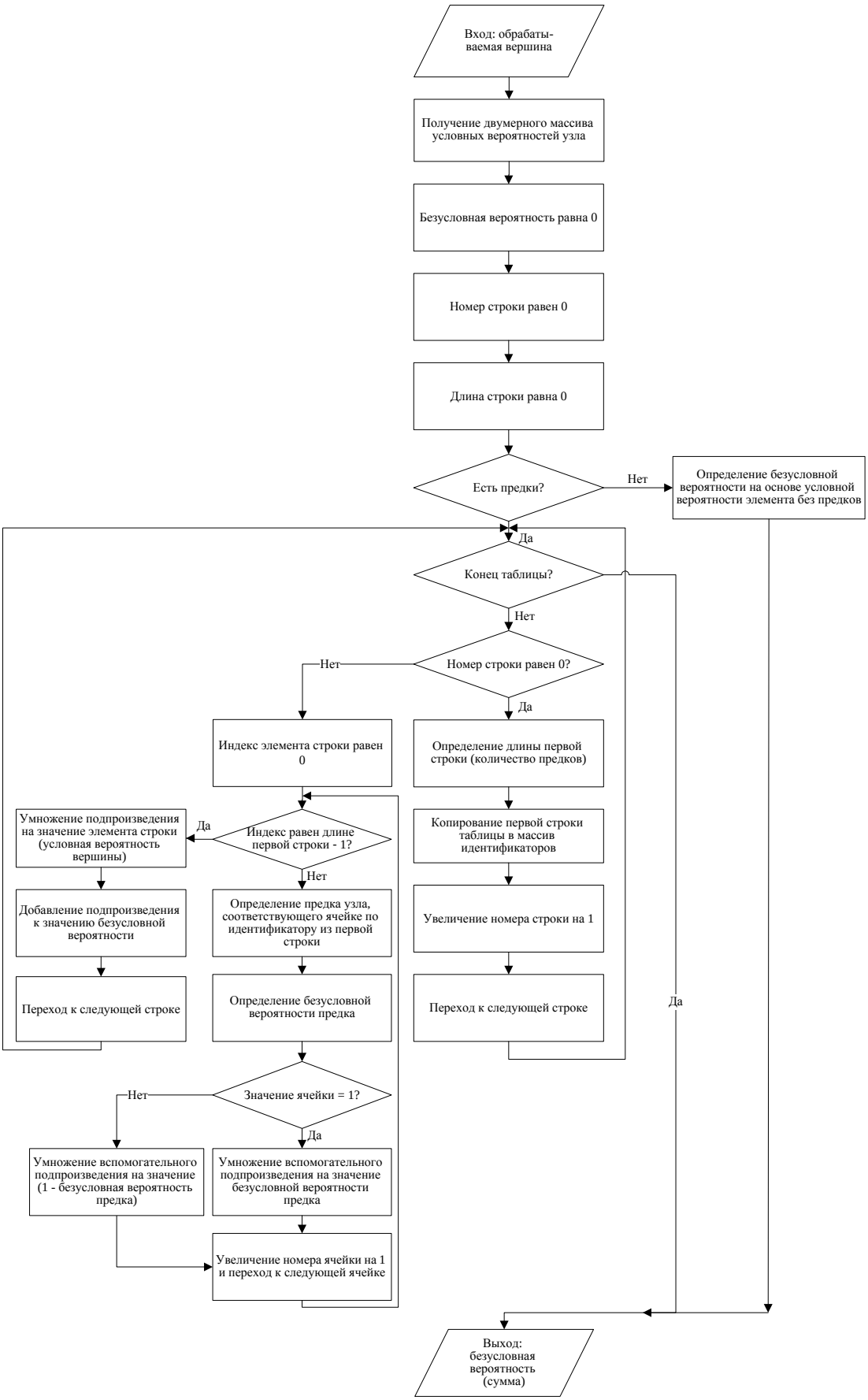


Схема функции определения безусловной вероятности вершины



Приложение Г – Пример вычисления вероятности атаки

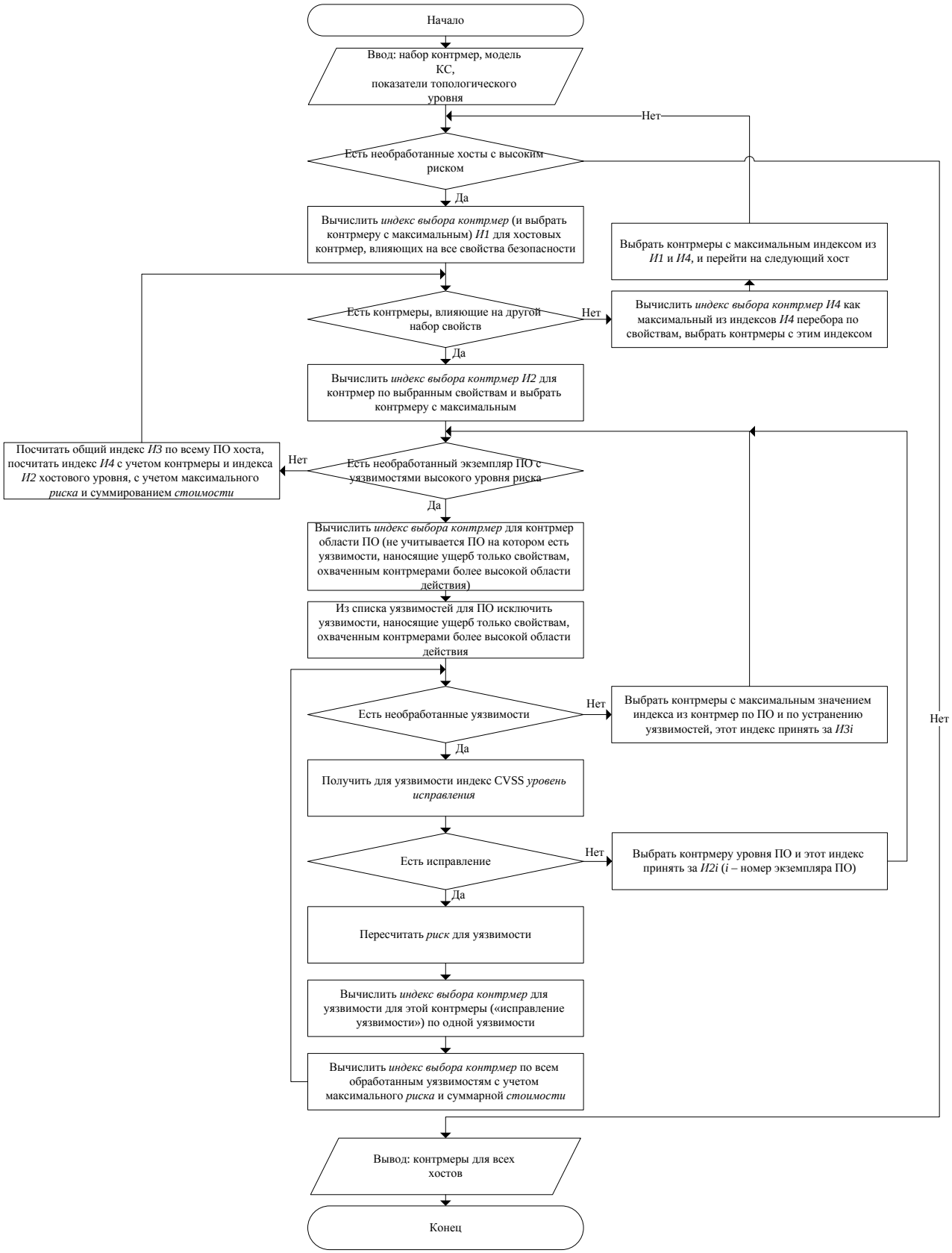
Вычисление вероятности компрометации узлов графа (рисунок 15) производится следующим образом:

1) Вычисление локальных вероятностей: для узла A (CVE-2006-0038) *AccessComplexity* «Medium», что соответствует численному значению 0,61, *Authentication* «None», что соответствует численному значению 0,704. Так как узел не является корневым, локальная вероятность рассчитывается по формуле $p(\text{A})=2\times$ $\times AccessComplexity\times Authentication=2\times0{,}61\times0{,}704=0{,}85888$; для узла C (CVE-2001-1572) *AccessVector* «Network» (численное значение 1), *AccessComplexity* «Low» (численное значение 0,71), *Authentication* «None» (численное значение 0,704). Так как узел является корневым (доступен с компьютера атакующего), локальная вероятность рассчитывается по формуле $p(\text{C})=2\times AccessVector\times AccessComplexity\times Authentication=2\times1\times0{,}71\times0{,}704=$ $=0{,}99968$; для узла B (CVE-2006-4572) *AccessVector* «Network» (численное значение 1), *AccessComplexity* «Low» (численное значение 0,71), *Authentication* «None» (численное значение 0,704). Так как узел является корневым, локальная вероятность рассчитывается по формуле $p(\text{B})=2\times AccessVector\times AccessComplexity\times Authentication=2\times1\times0{,}71\times0{,}704=$ $=0{,}99968$. Для узла D локальная вероятность может задаваться на уровне атакующего для определения вероятности начала атаки различными типами атакующих. Определим ее как 1.

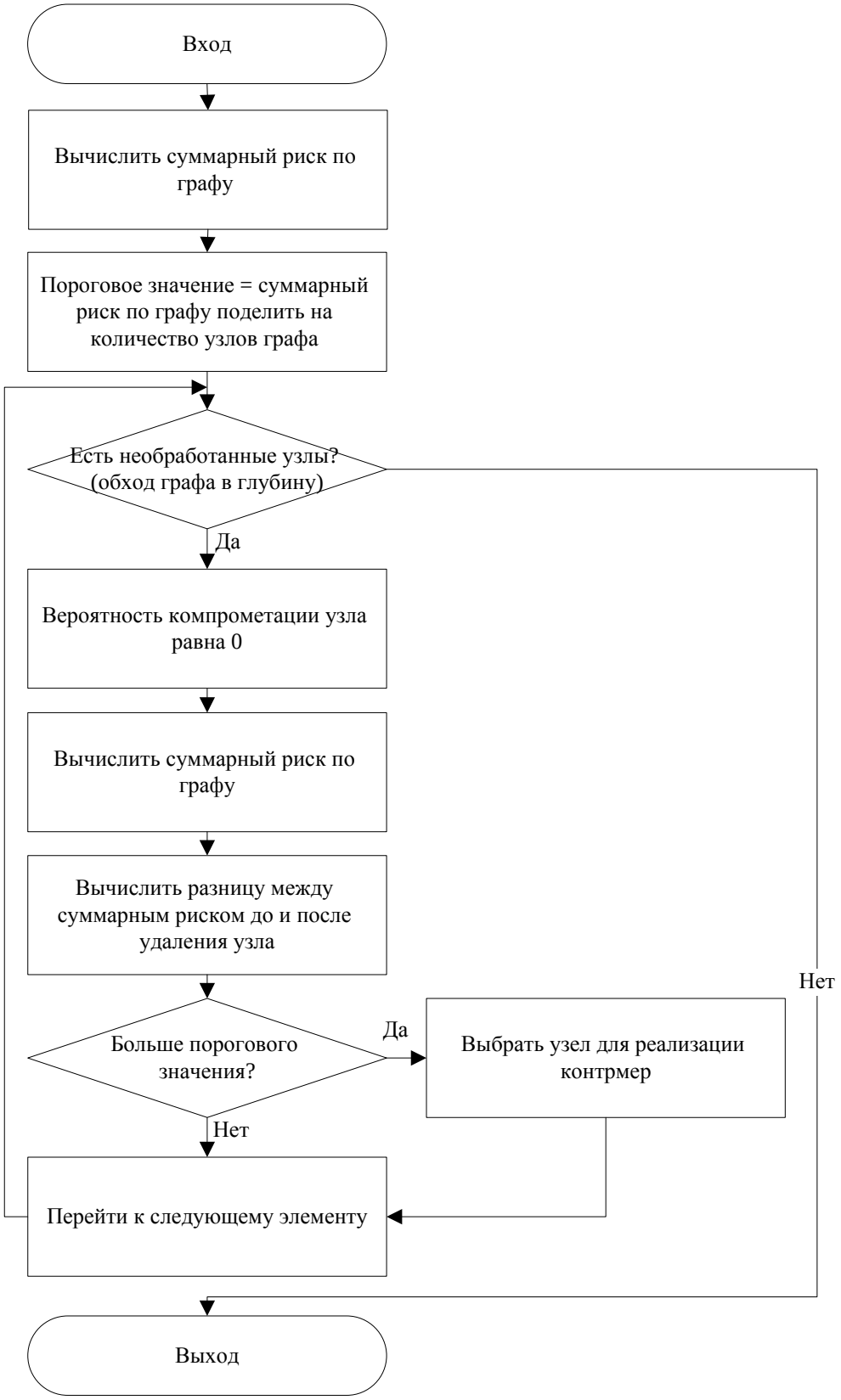
2) Вычисление условных вероятностей путем обратного обхода графа в глубину: для узла A условная вероятность определяется предками B и C, так как зависимость имеет тип «ИЛИ», $Pc(\text{A}|\text{B,C})=p(\text{A})=0{,}85888$ во всех случаях, кроме случая, когда B и C не скомпрометированы. Таблица дискретного локального распределения условных вероятностей для узла A представлена на рисунке 15, б; для узлов B и C условная вероятность определяется предком D, $Pc(\text{B}|\text{D})=p(\text{B})=0{,}99968$ и $Pc(\text{C}|\text{D})=p(\text{C})=0{,}99968$, если D скомпрометирован. Таблицы дискретного локального распределения условных вероятностей для узлов B и C представлены на рисунке 15, б.

3) Вычисление полных вероятностей путем маргинализации по известным вероятностям осуществляется путем обхода графа в ширину: для узла D $Pr(\text{D})=1$; для узла C $Pr(\text{C}=True\,|\,\text{D})=\sum_{D=\{True,False\}} Pc(\text{C}=True\,|\,\text{D})\times Pr(\text{D})=0{,}99968\times1+0\times0$; для узла B $Pr(\text{B}=True\,|\,\text{D})=\sum_{D=\{True,False\}} Pc(\text{B}=True\,|\,\text{D})\times Pr(\text{D})=0{,}99968\times1+0\times0$; для узла A $Pr(\text{A}=True\,|\,\text{B,C})=\sum_{\text{B,C}=\{True,False\}} Pc(\text{A}=True\,|\,\text{B,C})\times Pr(\text{B})\times Pr(\text{C})=0{,}85888\times0{,}99968\times$ $\times0{,}99968+0{,}85888\times0{,}99968\times0{,}00032+0{,}85888\times0{,}99968\times0{,}00032\approx0{,}86$.

Приложение Д – Схема алгоритма выбора контрмер на топологическом уровне



Приложение Е – Схема алгоритма определения узлов графа, представляющих наибольший риск



Приложение Ж – Классификация значений поля базы CAPEC

Таблица Ж.1 – Классификация значений поля *Indicators-Warnings of Attack* базы CAPEC

Значения поля *Indicators-Warnings of Attack*
Проникновение
If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
An attack designed to leverage a buffer overflow and redirect execution as per the attackers' bidding is fairly difficult to detect. An attack aimed solely at bringing the system down is usually preceded by a barrage of long inputs that make no sense. In either case, it is likely that the attacker would have resorted to a few hit-or-miss attempts that will be recorded in the system event logs, if they exist. -
Repeated submissions of incorrect secret values may indicate a brute force attack. For example, repeated bad passwords when accessing user accounts or repeated queries to databases using non-existent keys.
Attempts to download files protected by secrets (usually using encryption) may be a precursor to an offline attack to break the file's encryption and read its contents. This is especially significant if the file itself contains other secret values, such as password files.
If the attacker is able to perform the checking offline then there will likely be no indication that an attack is ongoing.
An example of indicator is when the client software crashes after executing code downloaded from a hostile server.
Many invalid login attempts are coming from the same machine (same IP address) or for the same log in name. The login attempts use passwords that are dictionary words.
Many exceptions are thrown by the application's filter modules in a short period of time. Check the logs. See if the probes are coming from the same IP address.
Differences in requests processed by the two entities. This requires careful monitoring or a capable log analysis tool.
The only indicators are multiple responses to a single request in the web logs. However, this is difficult to notice in the absence of an application filter proxy or a log analyzer. There are no indicators for the client,
Many incorrect login attempts are detected by the system.
Many incorrect attempts to answer the security question.
Null characters are observed by the filter. The filter needs to be able to understand various encodings of the Null character, or only canonical data should be passed to it.
There are no indicators for the server since a fixated session identifier is similar to an ordinarily generated one. However, too many invalid sessions due to invalid session identifiers is a potential warning.
A client can be suspicious if a received link contains preset session identifiers. However, this depends on the client's knowledge of such an issue. Also, fixation through Cross Site Scripting or hidden form fields is usually difficult to detect.
If the first decoding process has left some invalid or blacklisted characters, that may be a sign that the request is malicious.
Traffic filtering with IDS (or proxy) can detect requests with suspicious URLs. IDS may use signature based identification to reveal such URL based attacks.
The only indicators of successful Blind SQL Injection are the application or database logs that show similar queries with slightly differing logical conditions that increase in complexity over time. However, this requires extensive logging as well as knowledge of the queries that can be used to perform such injection and return meaningful information from the database.
Unicode encoded data is passed to APIs where it is not expected.
A web page that contains overly long UTF-8 codes constitute a protocol anomaly, and could be an indication that an attacker is attempting to exploit a vulnerability on the target host.
An attacker can use a fuzzer in order to probe for a UTF-8 encoding vulnerability. The fuzzer should generate suspicious network activity noticeable by an intrusion detection system.
An IDS filtering network traffic may be able to detect illegal UTF-8 characters.
Bad data is passed to the XML parser (possibly repeatedly), possibly making it crash or execute arbitrary code.
Bad data is passed to the XML parser, possibly making it crash.
Too many exceptions generated by the application as a result of malformed queries.
An attacker creating or modifying Symbolic links is a potential signal of attack in progress.
An attacker deleting temporary files can also be a sign that the attacker is trying to replace legitimate resources with malicious ones.
The client software crashes after visiting a URL downloaded from a hostile server.
Too many false or invalid queries to the database, especially those caused by malformed input.

Продолжение таблицы Ж.1

Значения поля *Indicators-Warnings of Attack*
Проникновение
An attacker can use a fuzzer in order to probe for this vulnerability. The fuzzer should generate suspicious network activity noticeable by an intrusion detection system.
If the first path decoding process has left some invalid or blacklisted characters, that may be a sign that the request is malicious.
Traffic filtering with IDS (or proxy) can detect request with suspicious URLs. IDS may use signature based identification to reveal such URL based attacks.
An attacker can use a fuzzer in order to probe for a UTF-8 encoding vulnerability. The fuzzer should generate suspicious network activity.
Too many exceptions generated by the application as a result of malformed XPath queries.
Bad data is continuously passed to the XML parser, possibly making it crash.
Эксплуатация
The log can have a trace of abnormal activity. Also if abnormal activity is detected on the host target. For instance flooding should be seen as abnormal activity and the target host may decide to take appropriate action in order to mitigate the attack (data filtering or blocking). Resource exhaustion is also a sign of abnormal activity.
A web penetration tool probing a web server may generate abnormal activities recorded on log files. Abnormal traffic such as a high number of request coming from the same client may also rise the warnings from a monitoring system or an intrusion detection tool.
Bad data is passed to the XML parser (possibly repeatedly), possibly making it crash or execute arbitrary code.
Bad data is passed to the XML parser, possibly making it crash.
Too many exceptions generated by the application as a result of malformed queries.
An attacker creating or modifying Symbolic links is a potential signal of attack in progress.
An attacker deleting temporary files can also be a sign that the attacker is trying to replace legitimate resources with malicious ones.
The client software crashes after visiting a URL downloaded from a hostile server.
Too many false or invalid queries to the database, especially those caused by malformed input.
An attacker can use a fuzzer in order to probe for this vulnerability. The fuzzer should generate suspicious network activity noticeable by an intrusion detection system.
If the first path decoding process has left some invalid or blacklisted characters, that may be a sign that the request is malicious.
Traffic filtering with IDS (or proxy) can detect request with suspicious URLs. IDS may use signature based identification to reveal such URL based attacks.
An attacker can use a fuzzer in order to probe for a UTF-8 encoding vulnerability. The fuzzer should generate suspicious network activity.
Too many exceptions generated by the application as a result of malformed XPath queries.
Bad data is continuously passed to the XML parser, possibly making it crash.
Разведка
A lot of invalid data is fed to the system. Data that cannot have been generated through a legitimate transaction/request. Data is coming into the system within a short period of time and potentially from the same IP.
Control characters are being detected by the filters repeatedly.
Repeated errors generated by the same piece of code are an indication, although it requires careful monitoring of the application and its associated error logs, if any.
You receive an e-mail from an entity that you are not even a customer of prompting you to log into your account.
You receive any e-mail that provides you with a link which takes you to a website on which you need to enter your log in information.

Приложение И – Данные для экспериментов

Топологии компьютерных сетей для экспериментов

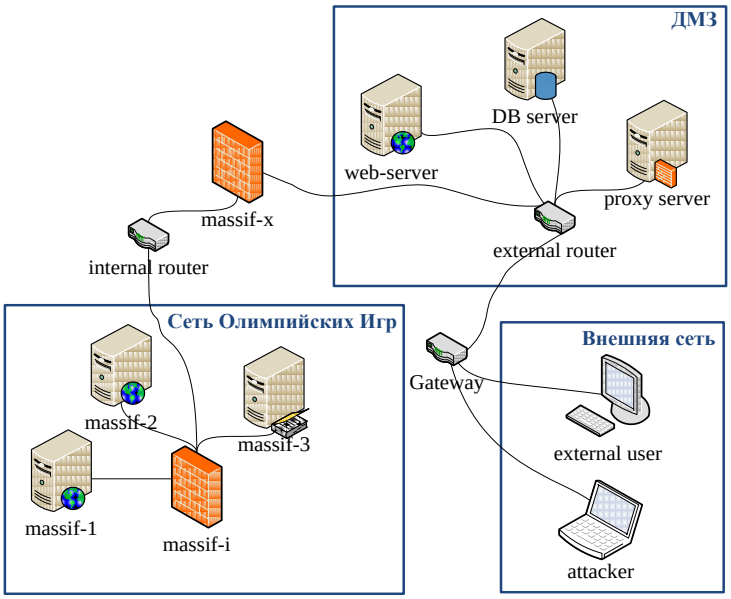


Рисунок И.1 – Топология сети, состоящей из 10 хостов

Для сети 1 критичными являются веб-приложения сети Олимпийских Игр (на хостах massif-1 и massif-2). Поэтому им присвоены значения критичности «Серьезная», или 10 по параметрам конфиденциальности, целостности и доступности: [10 10 10].

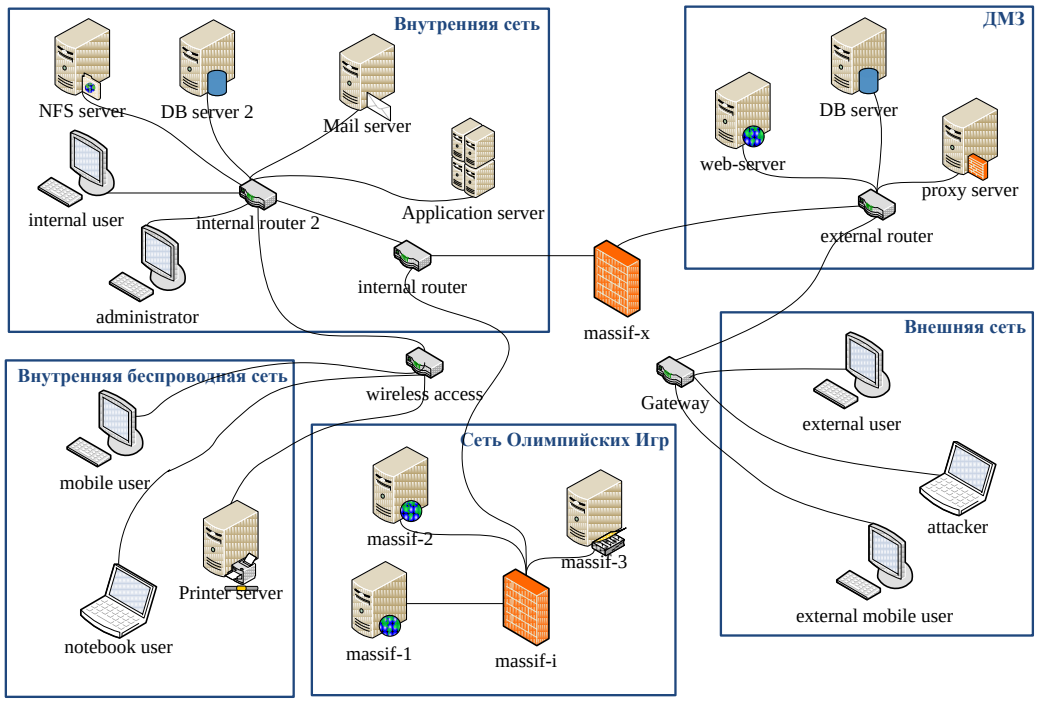


Рисунок И.2 – Топология сети, состоящей из 20 хостов

Для сети 2, помимо веб-приложений сети Олимпийских Игр критичным является почтовое приложение (на хосте *web-server* ДМЗ). Почтовому приложению сети 2 присвоено значение критичности «Повреждающая», или 1 по параметрам конфиденциальности, целостности и доступности: [1 1 1].

Для сети 3 были добавлены несколько критичных хостов: веб-сервер *web-server* ДМЗ с почтовым приложением (значение критичности «Повреждающая» – [1 1 1]); почтовый сервер *Mail server* внутренней сети (значение критичности «Значительная» – [0,3 0,3 0,3]); и рабочие станции внутренней сети *Workstation 2*, *Workstation 3*, *Workstation 4* (значение критичности «Значительная» – [0,5 0,5 0,5]).

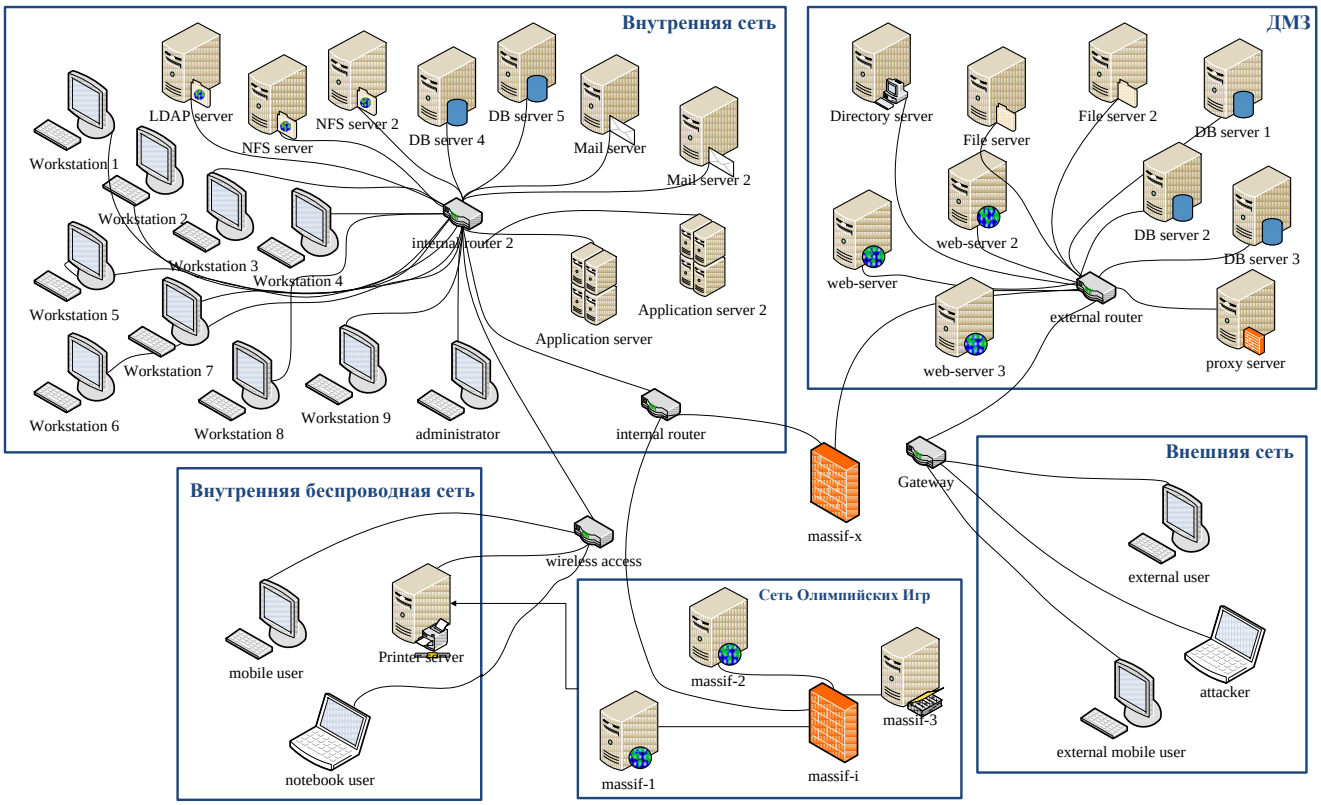


Рисунок И.3 – Топология сети, состоящей из 40 хостов

В таблице И.1 приведены названия и назначение хостов сети 1, сети 2 и сети 3.

Таблица И.1 – Хосты, входящие в состав тестовых сетей

Название	Назначение
massif-1, massif-2	веб-серверы сети Олимпийских Игр
massif-3	сервер аутентификации сети Олимпийских Игр
massif-i	брандмауэр сети Олимпийских Игр
massif-x	брандмауэр
web-server, web-server 2, web-server 3	веб-сервер ДМЗ
DB server, DB server 1, DB server 2, DB server 3, DB server 4, DB server 5	сервер баз данных
proxy server	прокси-сервер
Application server, Application server 2	сервер приложений
Mail server, Mail server 2	почтовый сервер

Продолжение таблицы И.1

Название	Назначение
NFS server, NFS server 2	NFS-сервер
File server, File server 2	Файловый сервер
Directory server	Сервер управления
LDAP server	LDAP-сервер
Workstation 1, Workstation 2, Workstation 3, Workstation 4, Workstation 5, Workstation 6, Workstation 7, Workstation 8, Workstation 9	Рабочие станции
external router	внешний маршрутизатор
internal router, internal router 2	внутренний маршрутизатор
external user	Персональный компьютер пользователя из внешней сети
Gateway	шлюз
Printer server	Сервер печати
attacker	Персональный компьютер атакующего из внешней сети
external mobile user	Мобильное устройство пользователя из внешней сети
mobile user	Мобильное устройство пользователя внутренней сети
notebook user	Портативный компьютер пользователя внутренней сети
administrator	Персональный компьютер администратора
internal user	Персональный компьютер пользователя внутренней сети

Зависимости сервисов для сетей, выбранных для экспериментов

Символ *W* на рисунке И.4 и рисунке И.5 обозначает вес соответствующей зависимости. Веб-приложения используют программную платформу ApacheStruts2 (доступ к веб-страницам осуществляется через порт 8080), поддерживаемую JBoss AS (порт 443). Для аутентификации используется NetIQ eDirect. Доступ к данным eDirect осуществляется через LDAP (Lightweight Directory Access Protocol), инкапсулированный в SSL (порт 636).

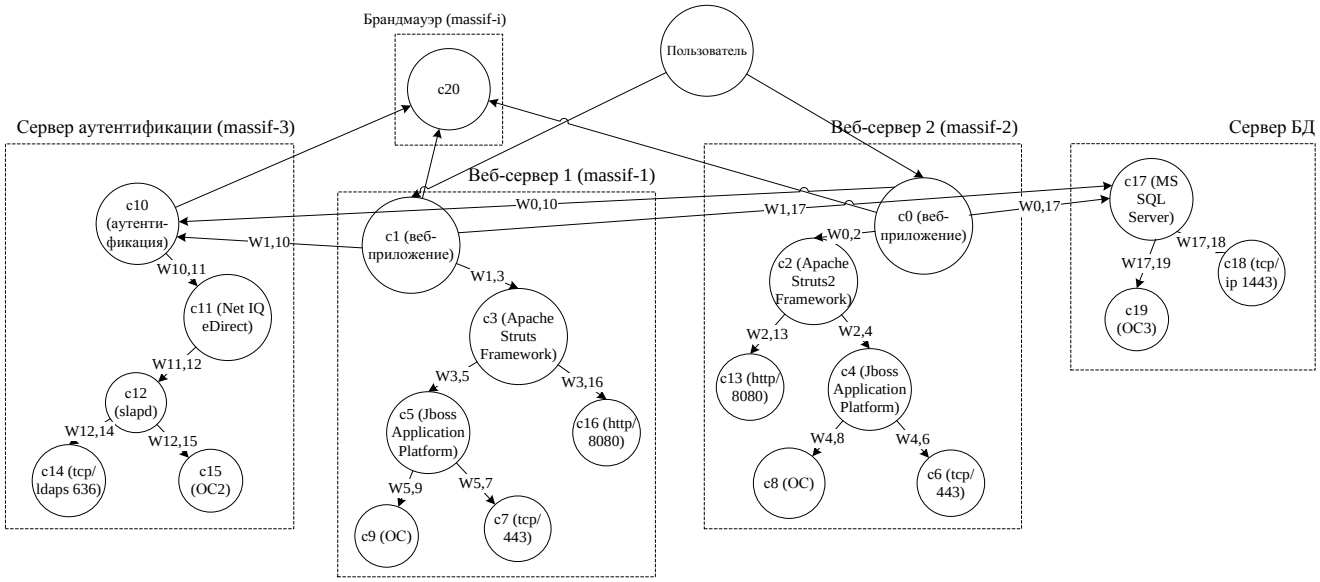


Рисунок И.4 – Зависимости сервисов для сети 1

Помимо этого, на рисунке И.5 удаленные пользователи получают доступ к почте через почтовое веб-приложение. Локальные пользователи могут использовать сервисы pops (порт 995) или imaps (порт 993). Для аутентификации используется SSL. Для доступа к файлам используется nfsd.

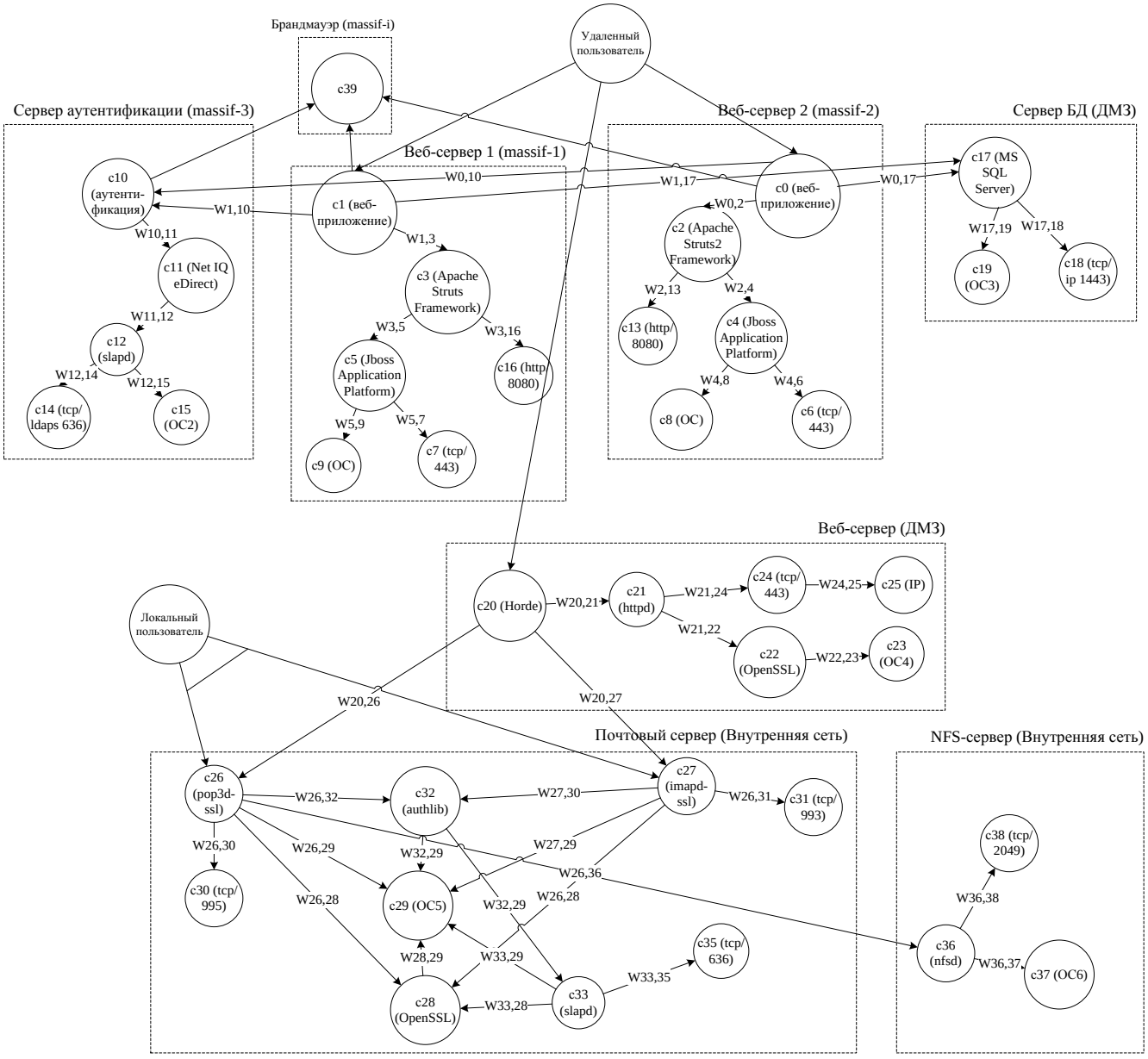


Рисунок И.5 – Зависимости сервисов для сети 2

Веса зависимостей сервисов и значения критичности сервисов для сетей, выбранных для экспериментов

Таблица И.2 – Веса зависимостей сервисов для сети 1

Сервисы	Apache-Struts2	JBoss AS	NetIQ eDirect	LDAP	порт 8080	порт 443	порт 636	сервис аутентификации	ОС
Веб-приложение	$\begin{bmatrix} 0,7 & 0,7 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$							$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,8 & 1 \end{bmatrix}$	
ApacheStruts2		$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$				
JBoss AS						$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$			$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
NetIQ eDirect				$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$					
LDAP							$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$		$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
сервис аутентификации			$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$						
SQL						$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$			$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Таблица И.3 – Итоговые значения критичности сервисов для сети 1

Сервис	Хост	Критичность
Веб-приложение (web application)	massif-1	[10 10 10]
ОС windows server 2008 (cpe:/o:microsoft:windows_server_2008::r2:x64)	massif-1	[10 10 10]
ApacheStruts2 (cpe:/a:apache:struts:2.0.0)	massif-1	[7 10 10]
JBoss AS (cpe:/a:redhat:jboss_community_application_server:5.0.1)	massif-1	[10 10 10]
port tcp/443	massif-1	[0 8 10]
port http/8080	massif-1	[0 8 10]
Веб-приложение (web application)	massif-2	[10 10 10]
ApacheStruts2 (cpe:/a:apache:struts:2.0.0)	massif-2	[7 10 10]
ОС windows server 2008 (cpe:/o:microsoft:windows_server_2008::r2:x64)	massif-2	[10 10 10]
port http/8080	massif-2	[0 8 10]
port tcp/443	massif-2	[0 8 10]

Продолжение Таблицы И.3

Сервис	Хост	Критичность
JBoss AS (cpe:/a:redhat:jboss_community_application_server:5.0.1)	massif-2	[10 10 10]
Сервис аутентификации (authentication service)	massif-3	[20 20 20]
ОС linux (cpe:/o:suse:linux_enterprise_server:9)	massif-3	[20 20 20]
port tcp/ldaps 636	massif-3	[0 16 20]
LDAP (slapd service)	massif-3	[20 20 20]
ОС linux (cpe:/o:linux:linux_kernel:2.6.27.33)	Сервер базы данных	[20 20 20]
SQL (cpe:/a:oracle:mysql:5.5.25)	Сервер базы данных	[20 20 20]
port tcp/443	Сервер базы данных	[20 20 20]
netFilter (cpe:/a:suse:suse_iptables)	massif-i	[20 20 20]
ОС linux (cpe:/o:suse:linux_enterprise_server:9)	massif-i	[20 20 20]
Citrix (cpe:/a:citrix:ica_client:6.1)	Брандмауэр	[20 20 20]
ОС linux (cpe:/o:linux:linux_kernel:2.6.27.33)	Брандмауэр	[20 20 20]

Таблица И.4 – Веса зависимостей сервисов для сети 2

Сервисы	httpd	imapd-ssl	pop3d-ssl	OpenSSL	slapd	порт 443	порт 995	порт 993	порт 636	порт 2049	nfsd	ОС
Horde	$\begin{bmatrix} 0,7 & 0,7 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,8 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,8 & 1 \end{bmatrix}$									
httpd				$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$		$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$						
OpenSSL												$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
pop3d-ssl				$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,8 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$		$\begin{bmatrix} 0,7 & 0,7 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$				$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,8 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,8 & 1 \end{bmatrix}$
imapd-ssl				$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$				$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
slapd									$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0,7 \\ 0 & 0,8 & 1 \end{bmatrix}$			$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
nfsd										$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,8 & 1 \end{bmatrix}$		$\begin{bmatrix} 0,7 & 0,7 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

Таблица И.5 – Итоговые значения критичности сервисов для сети 2

Сервис	**Хост**	**Критичность**
Веб-приложение (web application)	massif-1	[10 10 10]
ОС windows server 2008 (cpe:/o:microsoft:windows_server_2008::r2:x64)	massif-1	[10 10 10]
ApacheStruts2 (cpe:/a:apache:struts:2.0.0)	massif-1	[7 10 10]
JBoss AS (cpe:/a:redhat:jboss_community_application_server:5.0.1)	massif-1	[10 10 10]
port tcp/443	massif-1	[0 8 10]
port http/8080	massif-1	[0 8 10]
Веб-приложение (web application)	massif-2	[10 10 10]
ApacheStruts2 (cpe:/a:apache:struts:2.0.0)	massif-2	[7 10 10]
ОС windows server 2008 (cpe:/o:microsoft:windows_server_2008::r2:x64)	massif-2	[10 10 10]
port http/8080	massif-2	[0 8 10]
port tcp/443	massif-2	[0 8 10]
JBoss AS (cpe:/a:redhat:jboss_community_application_server:5.0.1)	massif-2	[10 10 10]
Сервис аутентификации (authentication service)	massif-3	[20 20 20]
ОС linux (cpe:/o:suse:linux_enterprise_server:9)	massif-3	[20 20 20]
port tcp/ldaps 636	massif-3	[0 16 20]
LDAP (slapd service)	massif-3	[20 20 20]
NetIQ eDirectory	massif-3	[20 20 20]
Horde (web mail)	web server	[1 1 1]
HTTP Server (httpd service)	web server	[0,7 1 1]
OpenSSL	web server	[1 1 1]
ОС windows server 2008 (cpe:/o:microsoft:windows_server_2008::r2:x64)	web server	[1 1 1]
port tcp/443	web server	[0 0,8 1]
ОС (cpe:/o:linux:linux_kernel:2.6.27.33)	Сервер базы данных	[20 20 20]
SQL (cpe:/a:oracle:mysql:5.5.25)	Сервер базы данных	[20 20 20]
port tcp/443	Сервер базы данных	[20 20 20]
netFilter (cpe:/a:suse:suse_iptables)	massif-i	[20 20 20]
ОС linux (cpe:/o:suse:linux_enterprise_server:9)	massif-i	[20 20 20]
OpenSSL	mail server	[1 1 1]
ОС linux (cpe:/o:novell:suse_linux:11:sp1:server)	mail server	[1 1 1]
Dovecot (imapd-ssl)	mail server	[1 1 1]
Postfix (pop3d-ssl service)	mail server	[1 1 1]
port tcp/ldaps 636	mail server	[0 0,8 1]
port tcp/993	mail server	[0 0,8 1]
port tcp/995	mail server	[0,7 1 1]
nfsd	NFS server	[1 1 1]
ОС linux (cpe:/o:linux:linux_kernel:2.6.27.33)	NFS server	[0,7 1 1]
port tcp/2049	NFS server	[1 1 1]
Citrix (cpe:/a:citrix:ica_client:6.1)	Брандмауэр	[20 20 20]
ОС linux (cpe:/o:linux:linux_kernel:2.6.27.33)	Брандмауэр	[20 20 20]

Приложение К – Отображение зависимостей сервисов для сети 2 и сети 3 в программном прототипе

Service links: web application (massif-1) - > apacheService (massif-1) AND web application (massif-1) - > authentication service (massif-3) AND web application (massif-1) - > sqlService (DB server) AND web application (massif-1) - > netFilterService (massif-i)
Service links: apacheService (massif-1) - > jbossService (massif-1) AND apacheService (massif-1) - > port http/8080 (massif-1)
Service links: jbossService (massif-1) - > windowsServerService (massif-1) AND jbossService (massif-1) - > port tcp/443 (massif-1)
Service links: web application (massif-2) - > apacheService (massif-2) AND web application (massif-2) - > authentication service (massif-3) AND web application (massif-2) - > sqlService (DB server) AND web application (massif-2) - > netFilterService (massif-i)
Service links: apacheService (massif-2) - > jbossService (massif-2) AND apacheService (massif-2) - > port http/8080 (massif-2)
Service links: jbossService (massif-2) - > windowsServerService (massif-2) AND jbossService (massif-2) - > port tcp/443 (massif-2)
Service links: authentication service (massif-3) - > eDirectoryService (massif-3) AND authentication service (massif-3) - > netFilterService (massif-i)
Service links: eDirectoryService (massif-3) - > slapd service (massif-3)
Service links: slapd service (massif-3) - > port tcp/ldaps 636 (massif-3) AND slapd service (massif-3) - > linuxSuseService (massif-3)
Service links: sqlService (DB server) - > linuxService (DB server) AND sqlService (DB server) - > port tcp/443 (DB server)
Service links: netFilterService (massif-i) - > linuxSuseService (massif-i)
Service links: hordeService (web-server) - > httpService (web-server) AND hordeService (web-server) - > postfixService (Mail server) AND hordeService (web-server) - > dovecotService (Mail server)
Service links: httpService (web-server) - > openSSLService (web-server) AND httpService (web-server) - > port tcp/443 (web-server)
Service links: openSSLService (web-server) - > windowsServer2008Service (web-server)
Service links: postfixService (Mail server) - > port tcp/995 (Mail server) AND postfixService (Mail server) - > openSSLService (Mail server) AND postfixService (Mail server) - > linuxSuseNovelService (Mail server) AND postfixService (Mail server) - > nfsd (NFS server) AND postfixService (Mail server) - > slapd service (Mail server)
Service links: dovecotService (Mail server) - > linuxSuseNovelService (Mail server) AND dovecotService (Mail server) - > openSSLService (Mail server) AND dovecotService (Mail server) - > port tcp/993 (Mail server) AND dovecotService (Mail server) - > slapd service (Mail server)
Service links: openSSLService (Mail server) - > linuxSuseNovelService (Mail server)
Service links: slapd service (Mail server) - > linuxSuseNovelService (Mail server) AND slapd service (Mail server) - > port tcp/636 (Mail server)
Service links: nfsd (NFS server) - > linuxService (NFS server) AND nfsd (NFS server) - > port tcp/2049 (FTP server)

Рисунок К.1 – Отображение зависимостей сервисов для сети 2 и сети 3 в программном прототипе

Приложение Л – Графы атакующих действий для сети 2 и сети 3 для атакующего 1

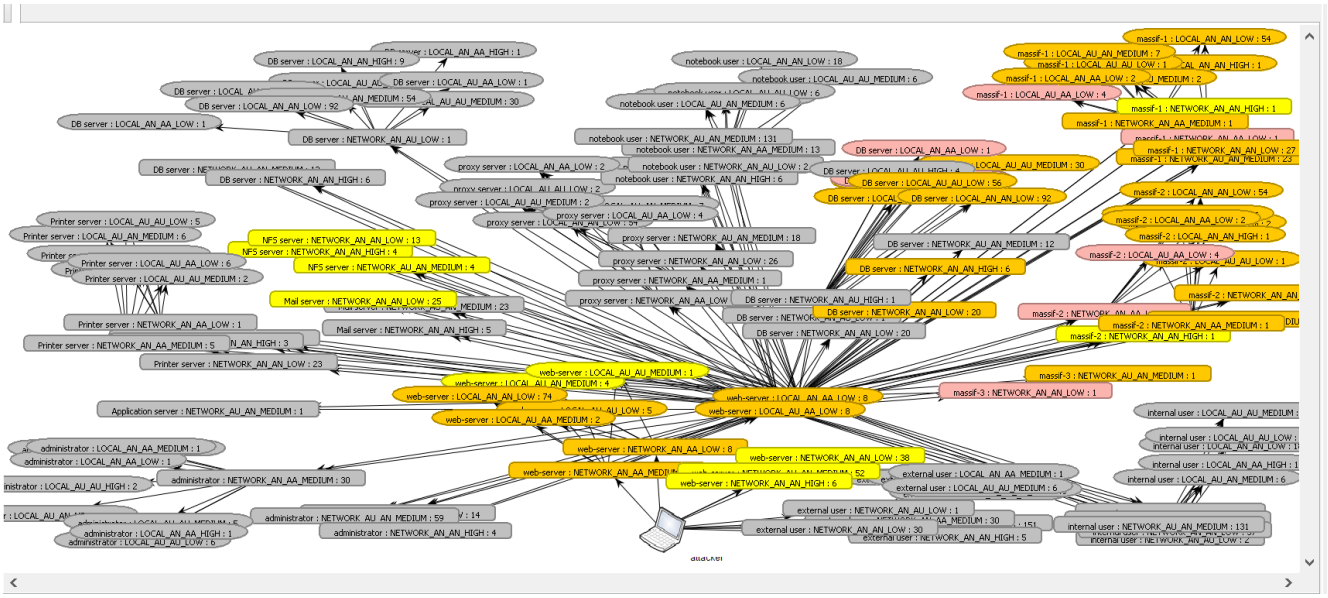


Рисунок Л.1 – Граф атакующих действий для сети 2 для атакующего 1

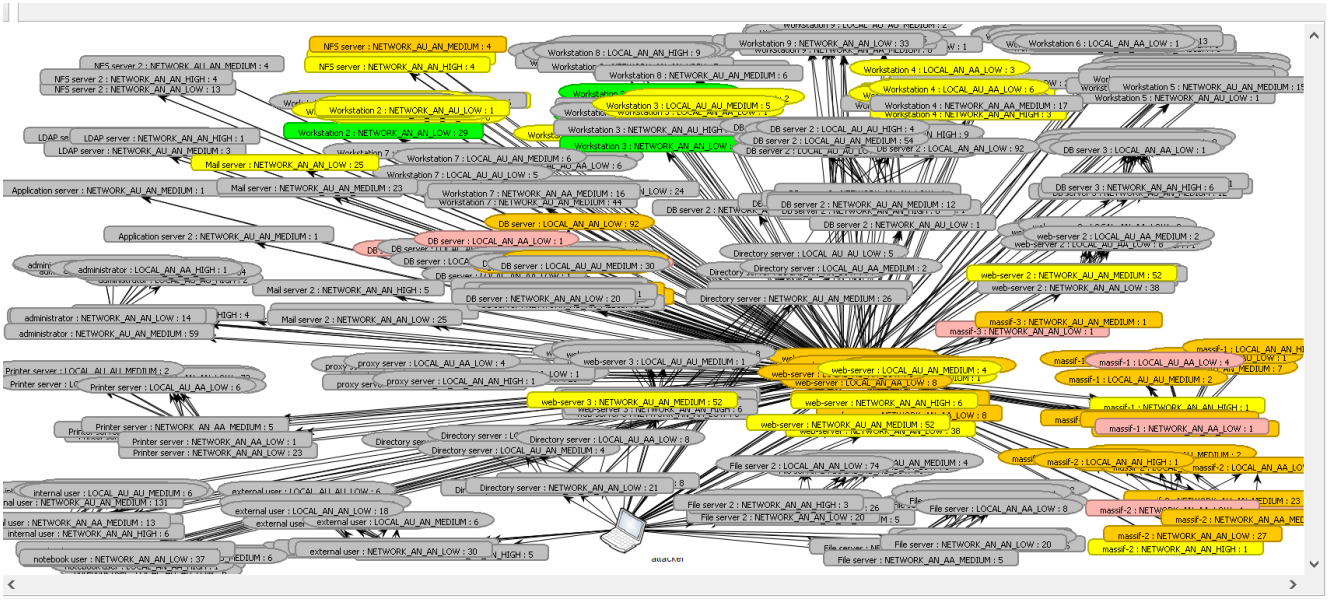


Рисунок Л.2 – Граф атакующих действий для сети 3 для атакующего 1

Приложение М – Значения риска для узлов графа атак сети 1 для атакующего 1

Таблица М.1 – Значения риска для узлов графа атак сети 1 для атакующего 1

Узел	Риск
massif-1: LOCAL_AN_AN_LOW	9,53
massif-1: NETWORK_AN_AN_LOW	1,48
massif-1: NETWORK_AN_AA_MEDIUM	9,18
massif-1: NETWORK_AN_AA_LOW	10,69
massif-1: NETWORK_AN_AN_HIGH	0,73
massif-1: NETWORK_AU_AN_MEDIUM	1,28
massif-1: LOCAL_AU_AU_LOW	3,95
massif-1: LOCAL_AN_AA_LOW	9,53
massif-1: LOCAL_AU_AA_LOW	11,86
massif-1: LOCAL_AN_AN_HIGH	1,57
massif-1: LOCAL_AU_AN_MEDIUM	2,73
massif-1: LOCAL_AU_AU_MEDIUM	3,4
massif-2: LOCAL_AU_AU_MEDIUM	3,4
massif-2: LOCAL_AU_AN_MEDIUM	2,73
massif-2: LOCAL_AN_AN_HIGH	1,57
massif-2: LOCAL_AN_AA_LOW	9,53
massif-2: LOCAL_AU_AA_LOW	11,86
massif-2: LOCAL_AN_AN_LOW	9,53
massif-2: NETWORK_AN_AA_MEDIUM	9,18
massif-2: NETWORK_AN_AA_LOW	10,69
massif-2: NETWORK_AN_AN_LOW	1,48
massif-2: NETWORK_AU_AN_MEDIUM	1,28
massif-2: LOCAL_AU_AU_LOW	3,95
massif-3: NETWORK_AN_AN_LOW	21,37
massif-3: NETWORK_AU_AN_MEDIUM	2,55
external user: NETWORK_AN_AN_LOW	0
external user: NETWORK_AN_AN_HIGH	0
external user: LOCAL_AU_AN_MEDIUM	0
external user: LOCAL_AU_AU_MEDIUM	0
external user: LOCAL_AN_AA_MEDIUM	0
external user: LOCAL_AN_AA_HIGH	0
external user: NETWORK_AN_AA_MEDIUM	0
external user: NETWORK_AU_AN_MEDIUM	0
external user: LOCAL_AU_AU_LOW	0
external user: LOCAL_AN_AN_LOW	0
DB server: LOCAL_AU_AU_LOW	3,34
DB server: LOCAL_AU_AU_HIGH	0
DB server: LOCAL_AN_AA_HIGH	8,26
DB server: LOCAL_AN_AN_HIGH	1,15
DB server: NETWORK_AN_AN_HIGH	1,46
DB server: NETWORK_AN_AU_HIGH	0
DB server: LOCAL_AU_AN_MEDIUM	14,4
DB server: LOCAL_AU_AU_MEDIUM	2,87
DB server: LOCAL_AN_AN_LOW	2,33
DB server: LOCAL_AN_AA_LOW	16,75
DB server: NETWORK_AN_AN_LOW	2,96
DB server: NETWORK_AU_AN_MEDIUM	0
DB server: NETWORK_AN_AU_LOW	0
web-server: LOCAL_AU_AA_LOW	0
web-server: LOCAL_AU_AU_MEDIUM	0
web-server: LOCAL_AU_AA_MEDIUM	0
web-server: LOCAL_AU_AN_MEDIUM	0

Продолжение таблицы М.1

Узел	Риск
web-server: NETWORK_AN_AN_LOW	0
web-server: NETWORK_AU_AN_MEDIUM	0
web-server: NETWORK_AN_AA_MEDIUM	0
web-server: NETWORK_AN_AA_LOW	0
web-server: LOCAL_AN_AA_LOW	0
web-server: NETWORK_AN_AN_HIGH	0
web-server: LOCAL_AU_AU_LOW	0
proxy server: NETWORK_AN_AN_LOW	0
proxy server: NETWORK_AN_AA_LOW	0
proxy server: LOCAL_AU_AU_LOW	0
proxy server: LOCAL_AN_AN_LOW	0
proxy server: LOCAL_AU_AA_LOW	0
proxy server: LOCAL_AN_AA_LOW	0
proxy server: LOCAL_AU_AN_MEDIUM	0
proxy server: LOCAL_AU_AU_MEDIUM	0
proxy server: NETWORK_AU_AN_MEDIUM	0
proxy server: NETWORK_AN_AA_MEDIUM	0

Из таблицы видно, что ненулевой риск соответствует узлам с ненулевой критичностью.

Приложение Н – Последовательности атак и событий безопасности для экспериментов

Таблица Н.1 – Последовательности атак и событий безопасности для экспериментов

Последовательность атакующих действий	**Последовательность событий безопасности**
Сеть 1	
Атакующий 1	
(1) Из внешней сети: CAPEC-286: Reconnaissance -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-408: Information Gathering from Traditional Sources -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-409: Information Gathering from Non-Traditional Sources -> (4) Атака на web-server: CAPEC-106: Cross Site Scripting through Log Files	Событие 1: web-server (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-170: Web Application Fingerprinting -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-224: Fingerprinting -> web-server: CAPEC-14: Client-side Injection-induced Buffer Overflow	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: web-server (CAPEC-14): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-312: Active OS Fingerprinting -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-313: Passive OS Fingerprinting -> (4) Атака на web-server: CAPEC-24: Filter Failure through Buffer Overflow -> (5) Атака на massif-2: CAPEC-63: Simple Script Injection	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: web-server (CAPEC-24): Many exceptions are thrown by the application's filter modules in a short period of time. Check the logs. See if the probes are coming from the same IP address.
(1) Из внешней сети: CAPEC-314: IP Fingerprinting Probes -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-315: TCP/IP Fingerprinting Probes -> (4) Атака на web-server: CAPEC-24: Filter Failure through Buffer Overflow -> (5) Атака на massif-2: CAPEC-64: Using Slashes and URL Encoding Combined to Bypass Validation Logic	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: web-server (CAPEC-24): Many exceptions are thrown by the application's filter modules in a short period of time. Check the logs. See if the probes are coming from the same IP address. Событие 3: massif-2 (CAPEC-64): If the first decoding process has left some invalid or blacklisted characters, that may be a sign that the request is malicious., Traffic filtering with IDS (or proxy) can detect requests with suspicious URLs. IDS may use signature based identification to reveal such URL based attacks.
(1) Из внешней сети: CAPEC-291: DNS Zone Transfers -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-292: Host Discovery -> (4) Атака на web-server: CAPEC-244: Cross-Site Scripting via Encoded URI Schemes -> (5) Атака на massif-2: CAPEC-101: Server Side Include (SSI) Injection	Событие 1: web-server (CAPEC-10): *CAPEC 10 description*
CAPEC 10 name: Buffer Overflow via Environment Variables *CAPEC 10 description*: If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.	

Продолжение таблицы Н.1

Последовательность атакующих действий	**Последовательность событий безопасности**
Сеть 1	
Атакующий 1	
(1) Из внешней сети: TCP Timestamp Probe-> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-321: TCP Sequence Number Probe -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на massif-1: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: massif-1 (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-405: Social Information Gathering via Research -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-406: Social Information Gathering via Dumpster Diving -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> proxy server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: proxy server (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-409: Information Gathering from Non-Traditional Sources -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-412: Pretexting via Customer Service -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на massif-3: CAPEC-106: Cross Site Scripting through Log Files	Событие 1: web-server (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-529: Malware-Directed Internal Reconnaissance -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-541: Application Fingerprinting (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на massif-2: CAPEC-10: *CAPEC 10 name* -> (6) Сбор информации с massif-2: CAPEC-85: AJAX Fingerprinting -> (7) Атака на massif-2: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: massif-2 (CAPEC-10): *CAPEC 10 description* Событие 3: massif-2 (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-472: Browser Fingerprinting -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-529: Malware-Directed Internal Reconnaissance -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на DB server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: *CAPEC 10 description*
(1) Из внешней сети: CAPEC-300: Port Scanning -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-301: TCP Connect Scan -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на massif-1: CAPEC-10: *CAPEC 10 name* -> (6) Сбор информации с massif-1: CAPEC-302: TCP FIN scan -> (7) Атака на massif-1: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: massif-1 (CAPEC-10): *CAPEC 10 description* Событие 3: massif-1 (CAPEC-10): *CAPEC 10 description*
CAPEC 10 name: Buffer Overflow via Environment Variables *CAPEC 10 description*: If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.	

Продолжение таблицы Н.1

Последовательность атакующих действий	**Последовательность событий безопасности**
Сеть 1	
Атакующий 1	
(1) Из внешней сети: CAPEC-299: TCP SYN Ping -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* ->
(3) Сбор информации с web-server: CAPEC-300: Port Scanning ->
(4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls ->
(5) Атака на DB server: CAPEC-10 *CAPEC 10 name* ->
(6) Сбор информации с DB server: CAPEC-301: TCP Connect Scan ->
(7) Атака на DB server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description*
Событие 2: DB server (CAPEC-10): *CAPEC 10 description*
Событие 3: DB server (CAPEC-10): *CAPEC 10 description*
Атакующий 2	
(1) Из внешней сети: CAPEC-286: Reconnaissance ->
(2) Атака на web-server: web-server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-85: AJAX Fingerprinting ->
(2) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls ->
(3) Сбор информации с web-server: CAPEC-286: Reconnaissance ->
(4) Атака на proxy server: CAPEC-14: Client-side Injection-induced Buffer Overflow ->
(5) Сбор информации с proxy server: CAPEC-378: WASC-45 – Fingerprinting ->
(6) Атака на proxy server: CAPEC-45: Buffer Overflow via Symbolic Links	Событие 1: proxy server (CAPEC-14): An example of indicator is when the client software crashes after executing code downloaded from a hostile server.
Событие 2: proxy server (CAPEC-45): An attacker creating or modifying Symbolic links is a potential signal of attack in progress., An attacker deleting temporary files can also be a sign that the attacker is trying to replace legitimate resources with malicious ones.
(1) Из внешней сети: CAPEC-286: Reconnaissance ->
(2) Атака на web-server: CAPEC-10: *CAPEC 10 name* ->
(3) Сбор информации с web-server: CAPEC-378: WASC-45 – Fingerprinting ->
(4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls ->
(5) Атака на massif-1: CAPEC-106: Cross Site Scripting through Log Files	Событие 1: web-server (CAPEC-10): *CAPEC 10 description*
Атакующий 3	
(1) Из внешней сети: CAPEC-407: Social Information Gathering via Pretexting ->
(2) Атака на web-server: CAPEC-10: *CAPEC 10 name* ->
(3) Сбор информации с web-server: CAPEC-408: Information Gathering from Traditional Sources ->
(4) Атака на DB server: CAPEC-231: XML Oversized Payloads ->
(5) Сбор информации с DB server: CAPEC-409: Information Gathering from Non-Traditional Sources ->
(6) Атака на DB server: CAPEC-78: Using Escaped Slashes in Alternate Encoding	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
Событие 2: DB server (CAPEC-231): Bad data is passed to the XML parser (possibly repeatedly), possibly making it crash or execute arbitrary code.
Событие 3: DB server (CAPEC-78): An attacker can use a fuzzer in order to probe for this vulnerability. The fuzzer should generate suspicious network activity noticeable by an intrusion detection system.
CAPEC 10 name: Buffer Overflow via Environment Variables *CAPEC 10 description*: If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.	

Продолжение таблицы Н.1

Последовательность атакующих действий	**Последовательность событий безопасности**
Сеть 1	
Атакующий 3	
(1) Из внешней сети: CAPEC-326: TCP Initial Window Size Probe -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-327: TCP Options Probe -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на massif-2: CAPEC-76: Manipulating Input to File System Calls	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
(1) Из внешней сети: CAPEC-324: TCP (ISN) Sequence Predictability Probe -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-325: TCP Congestion Control Flag (ECN) Probe -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на proxy server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert. Событие 2: proxy server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
Сеть 2	
Атакующий 1	
(1) Из внешней сети: CAPEC-286: Reconnaissance -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-378: WASC-45 – Fingerprinting -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на DB server: CAPEC-10: *CAPEC 10 name* -> (6) Сбор информации с DB server: CAPEC-169: Footprinting -> (7) Атака на DB server: CAPEC-108: Command Line Execution through SQL Injection	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert. Событие 2: DB server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
(1) Из внешней сети: CAPEC-315: TCP/IP Fingerprinting Probes -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-316: ICMP Fingerprinting Probes -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на massif-1: CAPEC-106: Cross Site Scripting through Log Files	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
(1) Из внешней сети: CAPEC-323: TCP (ISN) Counter Rate Probe -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-324: TCP (ISN) Sequence Predictability Probe -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на NFS server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert. Событие 2: NFS server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
CAPEC 10 name: Buffer Overflow via Environment Variables	

Продолжение таблицы Н.1

Последовательность атакующих действий	**Последовательность событий безопасности**
Сеть 2	
Атакующий 2	
(1) Из внешней сети: CAPEC-287: TCP SYN Scan -> (2) Атака на web-server: CAPEC-45: Buffer Overflow via Symbolic Links	Событие 1: web-server (CAPEC-45): An attacker creating or modifying Symbolic links is a potential signal of attack in progress., An attacker deleting temporary files can also be a sign that the attacker is trying to replace legitimate resources with malicious ones.
(1) Из внешней сети: CAPEC-415: Pretexting via Phone -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-467: Cross Site Identification -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на DB server: CAPEC-106: Cross Site Scripting through Log Files	Событие 1: web-server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
(1) Из внешней сети: CAPEC-413: Pretexting via Tech Support -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-414: Pretexting via Delivery Person -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на Mail server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: Mail server (CAPEC-10): *CAPEC 10 description*
Атакующий 3	
(1) Из внешней сети: CAPEC-170: Web Application Fingerprinting -> (2) Атака на web-server: CAPEC-66: SQL Injection -> (3) Сбор информации с web-server: CAPEC-224: Fingerprinting -> (4) Атака на massif-2: CAPEC-76: Manipulating Input to File System Calls	Событие 1: web-server (CAPEC-66): Too many false or invalid queries to the database, especially those caused by malformed input.
(1) Из внешней сети: CAPEC-300: Port Scanning -> (2) Атака на web-server: CAPEC-139: Relative Path Traversal -> (3) Сбор информации с web-server: CAPEC-301: TCP Connect Scan -> (4) Атака на massif-2: CAPEC-66: SQL Injection -> (5) Сбор информации с massif-2: CAPEC-302: TCP FIN scan -> (6) Атака на massif-2: CAPEC-78: Using Escaped Slashes in Alternate Encoding	Событие 1: massif-2 (CAPEC-66): Too many false or invalid queries to the database, especially those caused by malformed input. Событие 2: massif-2 (CAPEC-78): An attacker can use a fuzzer in order to probe for this vulnerability. The fuzzer should generate suspicious network activity noticeable by an intrusion detection system.
(1) Из внешней сети: CAPEC-308: UDP Scan -> (2) Атака на web-server: CAPEC-10: *CAPEC 10 name* -> (3) Сбор информации с web-server: CAPEC-309: Network Topology Mapping -> (4) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls -> (5) Атака на proxy server: CAPEC-10: *CAPEC 10 name* -> (6) Сбор информации с proxy server: CAPEC-311: OS Fingerprinting -> (7) Атака на proxy server: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server (CAPEC-10): *CAPEC 10 description* Событие 2: proxy server (CAPEC-10): *CAPEC 10 description* Событие 3: proxy server (CAPEC-10): *CAPEC 10 description*
CAPEC 10 name: Buffer Overflow via Environment Variables *CAPEC 10 description*: If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.	

Продолжение таблицы Н.1

Последовательность атакующих действий	**Последовательность событий безопасности**
Сеть 3	
Атакующий 1	
(1) Из внешней сети: CAPEC-378: WASC-45 – Fingerprinting -> (2) Атака на DB server 2: CAPEC-100: Overflow Buffers	Событие 1: DB server 2 (CAPEC-100): An attack designed to leverage a buffer overflow and redirect execution as per the attackers' bidding is fairly difficult to detect. An attack aimed solely at bringing the system down is usually preceded by a barrage of long inputs that make no sense. In either case, it is likely that the attacker would have resorted to a few hit-or-miss attempts that will be recorded in the system event logs, if they exist.
(1) Из внешней сети: CAPEC-319: IP (DF) 'Don't Fragment Bit' Echoing Probe -> (2) Атака на web-server 2: CAPEC-42: MIME Conversion -> (3) Сбор информации с web-server 2: CAPEC-320: TCP Timestamp Probe -> (4) Атака на web-server 2: CAPEC-24: Filter Failure through Buffer Overflow	Событие 1: web-server 2 (CAPEC-24): Many exceptions are thrown by the application's filter modules in a short period of time. Check the logs. See if the probes are coming from the same IP address.
(1) Из внешней сети: CAPEC-331: ICMP IP Total Length Field Probe -> (2) Атака на File server 2: CAPEC-24: Filter Failure through Buffer Overflow	Событие 1: File server 2 (CAPEC-24): Many exceptions are thrown by the application's filter modules in a short period of time. Check the logs. See if the probes are coming from the same IP address.
(1) Из внешней сети: CAPEC-287: TCP SYN Scan ->
(2) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls ->
(3) Атака на web-server: CAPEC-290: Enumerate Mail Exchange (MX) Records ->
(4) Атака на DB server: CAPEC-10: Buffer Overflow via Environment Variables	Событие 1: DB server (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
(1) Из внешней сети: CAPEC-407: Social Information Gathering via Pretexting ->
(2) Атака на web-server: CAPEC-76: Manipulating Input to File System Calls ->
(3) Сбор информации с web-server: CAPEC-408: Information Gathering from Traditional Sources ->
(4) Атака на Workstation 7: CAPEC-10: Buffer Overflow via Environment Variables	Событие 1: Workstation 7 (CAPEC-10): If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.
Атакующий 2	
(1) Из внешней сети: CAPEC-322: TCP (ISN) Greatest Common Divisor Probe ->
(2) Атака на web-server 3: CAPEC-10: *CAPEC 10 name* ->
(3) Сбор информации с web-server 3: CAPEC-323: TCP (ISN) Counter Rate Probe ->
(4) Атака на web-server 3: CAPEC-10: *CAPEC 10 name*	Событие 1: web-server 3 (CAPEC-10): *CAPEC 10 description*
Событие 2: web-server 3 (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-326: TCP Initial Window Size Probe ->
(2) Атака на Directory server: CAPEC-63: Simple Script Injection ->
(3) Сбор информации с Directory server: CAPEC-327: TCP Options Probe ->
(4) Атака на Directory server: CAPEC-14: *CAPEC 14 name*	Событие 1: Directory server (CAPEC-14): An example of indicator is when the client software crashes after executing code downloaded from a hostile server.

Продолжение таблицы Н.1

Последовательность атакующих действий	**Последовательность событий безопасности**
Сеть 3	
Атакующий 2	
(1) Из внешней сети: CAPEC-300: Port Scanning (2) Атака на web-server: CAPEC-76: *CAPEC 76 name* ->> (3) Сбор информации с web-server: CAPEC-301: TCP Connect Scan ->> (4) Атака на DB server: CAPEC-10: *CAPEC 10 name* ->> (5) Сбор информации с DB server: CAPEC-302: TCP FIN scan ->> (6) Атака на DB server: CAPEC-76: *CAPEC 76 name*	Событие 1: DB server (CAPEC-10): *CAPEC 10 description*
Атакующий 3	
(1) Из внешней сети: CAPEC-286: Reconnaissance ->> (2) Атака на DB server 3: CAPEC-10: *CAPEC 10 name* ->> (3) Сбор информации с DB server 3: CAPEC-378: WASC-45 – Fingerprinting ->> (4) Атака на DB server 3: CAPEC-76: *CAPEC 76 name*	Событие 1: DB server 3 (CAPEC-10): *CAPEC 10 description*
(1) Из внешней сети: CAPEC-313: Passive OS Fingerprinting ->> (2) Атака на web-server 3: CAPEC-86: Embedding Script (XSS) in HTTP Headers ->> (3) Сбор информации с web-server 3: CAPEC-314: IP Fingerprinting Probes ->> (4) Атака на web-server 3: CAPEC-14: *CAPEC 14 name*	Событие 1: web-server 3 (CAPEC-14): An example of indicator is when the client software crashes after executing code downloaded from a hostile server.
(1) Из внешней сети: CAPEC-170: Web Application Fingerprinting ->> (2) Атака на web-server: CAPEC-106: Cross Site Scripting through Log Files ->> (3) Сбор информации с web-server: CAPEC-224: Fingerprinting ->> (4) Атака на web-server: CAPEC-14: *CAPEC 14 name* ->> (5) Атака на Workstation 5: CAPEC-76: *CAPEC 76 name* ->> (6) Сбор информации с Workstation 5: CAPEC-285: ICMP Echo Request Ping ->> (7) Атака на Workstation 5: CAPEC-64: Using Slashes and URL Encoding Combined to Bypass Validation Logic	Событие 1: web-server (CAPEC-14): An example of indicator is when the client software crashes after executing code downloaded from a hostile server. Событие 2: Workstation 5 (CAPEC-64): If the first decoding process has left some invalid or blacklisted characters, that may be a sign that the request is malicious., Traffic filtering with IDS (or proxy) can detect requests with suspicious URLs. IDS may use signature based identification to reveal such URL based attacks.
CAPEC 76 name: Manipulating Input to File System Calls *CAPEC 10 name:* Buffer Overflow via Environment Variables *CAPEC 14 name:* Client-side Injection-induced Buffer Overflow *CAPEC 10 description*: If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.	

Приложение О – Изменение значений риска узлов графа атак для тестовой последовательности атаки и событий

Рассмотрим подробнее изменение значений риска компрометации узлов графа лля сети 1, граф атак которой изображен на рисунке 33, для последовательности атаки: CAPEC-529: Malware-Directed Internal Reconnaissance из внешней сети -> CAPEC-10: Buffer Overflow via Environment Variables на хосте web-server -> CAPEC-541: Application Fingerprinting с хоста web-server -> CAPEC-10: Buffer Overflow via Environment Variables на хосте DB server -> CAPEC-85: AJAX Fingerprinting с хоста DB server -> CAPEC-10: Buffer Overflow via Environment Variables на хосте DB server. И последовательности событий: Событие 1 (хост web-server, шаблон атаки CAPEC-10) «If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.» -> Событие 2 (хост DB server, шаблон атаки CAPEC-10) «If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.» -> Событие 3 (хост DB server, шаблон атаки CAPEC-10) «If the application does bound checking, it should fail when the data source is larger than the size of the destination buffer. If the application's code is well written, that failure should trigger an alert.»

После поступления события 1 вероятность компрометации узла «web-server: LOCAL_AN_AA_LOW» меняется с 0,54 на 1. Тем не менее, поскольку критичность данного узла равна 0, риск для узла остается равным 0.

Для потомков данного узла вероятности и риски меняются следующим образом: для узла «DB server: NETWORK_AN_AU_HIGH» вероятность меняется с 0,27 на 0,32, поскольку критичность данного узла равна 0, риск для узла остается равным 0; для узла «DB server: NETWORK_AN_AU_LOW» вероятность меняется с 0,54 на 0,64, поскольку критичность данного узла равна 0, риск для узла остается равным 0; для узла «DB server: LOCAL_AN_AA_HIGH» вероятность меняется с 0,21 на 0,24, риск для узла меняется с 8,26 (уровень «Высокий») на 9,39 (уровень «Высокий»); для узла «DB server: LOCAL_AN_AN_LOW» вероятность меняется с 0,42 на 0,48, риск для узла меняется с 2,33 (уровень «Высокий») на 2,65 (уровень «Высокий»); для узла «DB server: LOCAL_AU_AU_LOW» вероятность меняется с 0,61 на 0,69, риск для узла меняется с

3,34 (уровень «Высокий») на 3,8 (уровень «Высокий»); для узла «DB server: LOCAL_AU_AU_HIGH» вероятность меняется с 0,26 на 0,3, риск для узла остается равным 0; для узла «DB server: LOCAL_AN_AA_LOW» вероятность меняется с 0,42 на 0,48, риск для узла меняется с 16,76 (уровень «Критический») на 19,05 (уровень «Критический»); для узла «DB server: LOCAL_AN_AN_HIGH» вероятность меняется с 0,21 на 0,24, риск для узла меняется с 1,15 (уровень «Высокий») на 1,3 (уровень «Высокий»); для узла «DB server: LOCAL_AU_AN_MEDIUM» вероятность меняется с 0,36 на 0,41, риск для узла меняется с 14,4 (уровень «Критический») на 16,36 (уровень «Критический»); для узла «DB server: LOCAL_AU_AU_MEDIUM» вероятность меняется с 0,52 на 0,59, риск для узла меняется с 2,87 (уровень «Высокий») на 3,26 (уровень «Высокий»); для узла «DB server: NETWORK_AU_AN_MEDIUM» вероятность меняется с 0,46 на 0,55, риск для узла остается равным 0; для узла «DB server: NETWORK_AN_AN_LOW» вероятность меняется с 0,54 на 0,64, риск для узла меняется с 2,97 (уровень «Высокий») на 3,52 (уровень «Высокий»); для узла «DB server: NETWORK_AN_AN_HIGH» вероятность меняется с 0,27 на 0,32, риск для узла меняется с 1,46 (уровень «Высокий») на 1,73 (уровень «Высокий»). Для узла «massif-1: NETWORK_AN_AA_MEDIUM» вероятность меняется с 0,46 на 0,55, риск для узла меняется с 9,18 (уровень «Критический») на 10,87 (уровень «Критический»); для узла «massif-1: NETWORK_AN_AA_LOW» вероятность меняется с 0,54 на 0,64, риск для узла меняется с 10,69 (уровень «Критический») на 12,65 (уровень «Критический»); для узла «massif-1: LOCAL_AU_AN_MEDIUM» вероятность меняется с 0,41 на 0,46, риск для узла меняется с 2,73 (уровень «Высокий») на 3,03 (уровень «Высокий»); для узла «massif-1: LOCAL_AU_AU_LOW» вероятность меняется с 0,6 на 0,67, риск для узла меняется с 3,95 (уровень «Высокий») на 4,39 (уровень «Высокий»); для узла «massif-1: LOCAL_AN_AA_LOW» вероятность меняется с 0,48 на 0,54, риск для узла меняется с 9,53 (уровень «Высокий») на 10,59 (уровень «Критический»); для узла «massif-1: LOCAL_AN_AN_LOW» вероятность меняется с 0,48 на 0,54, риск для узла меняется с 9,53 (уровень «Высокий») на 10,59 (уровень «Критический»); для узла «massif-1: LOCAL_AU_AA_LOW» вероятность меняется с 0,6 на 0,67, риск для узла меняется с 11,86 (уровень «Критический») на 13,18 (уровень «Критический»); для узла «massif-1: LOCAL_AU_AU_MEDIUM» вероятность меняется с 0,52 на 0,57, риск для узла меняется с 2,73 (уровень «Высокий») на 3,78 (уровень «Высокий»); для узла «massif-1:

LOCAL_AN_AN_HIGH» вероятность меняется с 0,24 на 0,26, риск для узла меняется с 1,57 (уровень «Высокий») на 1,74 (уровень «Высокий»); для узла «massif-1: NETWORK_AN_AN_HIGH» вероятность меняется с 0,27 на 0,32, риск для узла меняется с 0,73 (уровень «Средний») на 0,87 (уровень «Высокий»); для узла «massif-1: NETWORK_AN_AN_LOW» вероятность меняется с 0,54 на 0,64, риск для узла меняется с 1,484 (уровень «Высокий») на 1,757 (уровень «Высокий»); для узла «massif-1: NETWORK_AU_AN_MEDIUM» вероятность меняется с 0,46 на 0,55, риск для узла меняется с 1,28 (уровень «Высокий») на 1,51 (уровень «Высокий»). Для узлов хоста massif-2 изменения аналогичны. Для узла «massif-3: NETWORK_AN_AN_LOW» вероятность меняется с 0,54 на 0,64, риск для узла меняется с 21,37 (уровень «Критический») на 25,3 (уровень «Критический»); для узла «massif-3: NETWORK_AU_AN_MEDIUM» вероятность меняется с 0,46 на 0,55, риск для узла меняется с 2,55 (уровень «Высокий») на 3,02 (уровень «Высокий»).

После поступления события 2 вероятность компрометации узла «DB server: NETWORK_AN_AU_HIGH» меняется с 0,27 на 1, риск для узла остается равным 0. Для узла «DB server: LOCAL_AN_AA_HIGH» вероятность меняется с 0,21 на 0,32, риск для узла меняется с 8,26 (уровень «Высокий») на 12,47 (уровень «Критический»); для узла «DB server: LOCAL_AN_AN_LOW» вероятность меняется с 0,42 на 0,64, риск для узла меняется с 2,33 (уровень «Высокий») на 3,52 (уровень «Высокий»); для узла «DB server: LOCAL_AU_AU_LOW» вероятность меняется с 0,61 на 0,92, риск для узла меняется с 3,34 (уровень «Высокий») на 5,05 (уровень «Высокий»); для узла «DB server: LOCAL_AU_AU_HIGH» вероятность меняется с 0,26 на 0,39, риск для узла остается равным 0; для узла «DB server: LOCAL_AN_AA_LOW» вероятность меняется с 0,61 на 0,64, риск для узла меняется с 16,76 (уровень «Критический») на 25,3 (уровень «Критический»); для узла «DB server: LOCAL_AN_AN_HIGH» вероятность меняется с 0,21 на 0,32, риск для узла меняется с 1,15 (уровень «Высокий») на 1,73 (уровень «Высокий»); для узла «DB server: LOCAL_AU_AN_MEDIUM» вероятность меняется с 0,36 на 0,55, риск для узла меняется с 14,4 (уровень «Критический») на 21,74 (уровень «Критический»); для узла «DB server: LOCAL_AU_AU_MEDIUM» вероятность меняется с 0,52 на 0,79, риск для узла меняется с 2,87 (уровень «Высокий») на 4,34 (уровень «Высокий»).

Третье событие фиксирует компрометацию целевого узла.

Приложение П – Результаты экспериментов по определению выигрыша в случае реализации контрмер

Таблица П.1 – Результаты экспериментов по определению выигрыша в случае реализации контрмер

Последовательность атаки	**Событие**	**Потери до, €**	**Потери после, €**	**Индекс *AL*, €**
Сеть 1				
Атакующий 1				
Последовательность 1 (число шагов: 7)	Событие 1	264000	26400	237600 (90%)
	Событие 2	264000	224400	39600 (15%)
	Событие 3	264000	264000	0 (0%)
Последовательность 2 (число шагов: 6)	Событие 1	55000	14150	40850 (74%)
	Событие 2	55000	14150	40850 (74%)
	Событие 3	55000	55000	0 (0%)
Последовательность 3 (число шагов: 7)	Событие 1	396000	26400	369600 (93%)
	Событие 2	396000	224400	144600 (37%)
	Событие 3	396000	396000	0 (0%)
Последовательность 4 (число шагов: 7)	Событие 1	264000	26400	237600 (90%)
	Событие 2	264000	224400	39600 (15%)
	Событие 3	264000	264000	0 (0%)
Последовательность 5 (число шагов: 5)	Событие 1	396000	26400	369600 (93%)
	Событие 2	396000	224400	144600 (37%)
	Событие 3	396000	396000	0 (0%)
Последовательность 6 (число шагов: 5)	Событие 1	27500	0	27500 (100%)
	Событие 2	27500	0	27500 (100%)
	Событие 3	27500	27500	0 (0%)
Последовательность 7 (число шагов: 6)	Событие 1	0	0	0
	Событие 2	0	0	0
	Событие 3	0	0	0
Последовательность 8 (число шагов: 5)	Событие 1	198000	26400	171600 (87%)
	Событие 2	198000	198000	0 (0%)
	Событие 3	198000	198000	0 (0%)
Последовательность 9 (число шагов: 7)	Событие 1	396000	99400	296600 (75%)
	Событие 2	396000	99400	296600 (75%)
	Событие 3	396000	99400	296600 (75%)
Последовательность 10 (число шагов: 6)	Событие 1	55000	14150	40850 (74%)
	Событие 2	55000	14150	40850 (74%)
	Событие 3	55000	55000	0 (0%)
Атакующий 2				
Последовательность 1 (число шагов: 6)	Событие 1	264000	26400	237600 (90%)
	Событие 2	264000	224400	39600 (15%)
	Событие 3	264000	264000	0 (0%)
Последовательность 2 (число шагов: 7)	Событие 1	396000	26400	369600 (93%)
	Событие 2	396000	224400	144600 (37%)
	Событие 3	396000	396000	0 (0%)
Последовательность 3 (число шагов: 7)	Событие 1	0	0	0
	Событие 2	0	0	0
	Событие 3	0	0	0
Последовательность 4 (число шагов: 7)	Событие 1	396000	99400	296600 (75%)
	Событие 2	396000	99400	296600 (75%)
	Событие 3	396000	396000	0 (0%)
Последовательность 5 (число шагов: 7)	Событие 1	55000	14150	40850 (74%)
	Событие 2	55000	14150	40850 (74%)
	Событие 3	55000	55000	0 (0%)

Продолжение таблицы П.1

Последовательность атаки	**Событие**	**Потери до, €**	**Потери после, €**	**Индекс *AL*, €**
Сеть 1				
Атакующий 2				
Последовательность 6 (число шагов: 7)	Событие 1	198000	26400	171600 (87%)
	Событие 2	198000	198000	0 (0%)
	Событие 3	198000	198000	0 (0%)
Последовательность 7 (число шагов: 6)	Событие 1	55000	14150	40850 (74%)
	Событие 2	55000	14150	40850 (74%)
	Событие 3	55000	55000	0 (0%)
Последовательность 8 (число шагов: 7)	Событие 1	396000	26400	369600 (93%)
	Событие 2	396000	224400	144600 (37%)
	Событие 3	396000	396000	0 (0%)
Последовательность 9 (число шагов: 6)	Событие 1	264000	26400	237600 (90%)
	Событие 2	264000	224400	39600 (15%)
	Событие 3	264000	264000	0 (0%)
Последовательность 10 (число шагов: 7)	Событие 1	396000	26400	369600 (93%)
	Событие 2	396000	224400	144600 (37%)
	Событие 3	396000	396000	0 (0%)
Атакующий 3				
Последовательность 1 (число шагов: 5)	Событие 1	27500	0	27500 (100%)
	Событие 2	27500	0	27500 (100%)
	Событие 3	27500	27500	0 (0%)
Последовательность 2 (число шагов: 6)	Событие 1	396000	26400	369600 (93%)
	Событие 2	396000	224400	144600 (37%)
	Событие 3	396000	396000	0 (0%)
Последовательность 3 (число шагов: 5)	Событие 1	0	0	0
	Событие 2	0	0	0
	Событие 3	0	0	0
Последовательность 4 (число шагов: 6)	Событие 1	0	0	0
	Событие 2	0	0	0
	Событие 3	0	0	0
Последовательность 5 (число шагов: 5)	Событие 1	55000	14150	40850 (74%)
	Событие 2	55000	14150	40850 (74%)
	Событие 3	55000	55000	0 (0%)
Последовательность 6 (число шагов: 6)	Событие 1	396000	99400	296600 (75%)
	Событие 2	396000	99400	296600 (75%)
	Событие 3	396000	396000	0 (0%)
Последовательность 7 (число шагов: 5)	Событие 1	27500	0	27500 (100%)
	Событие 2	27500	0	27500 (100%)
	Событие 3	27500	27500	0 (0%)
Последовательность 8 (число шагов: 6)	Событие 1	396000	26400	369600 (93%)
	Событие 2	396000	224400	144600 (37%)
	Событие 3	396000	396000	0 (0%)
Последовательность 9 (число шагов: 5)	Событие 1	55000	14150	40850 (74%)
	Событие 2	55000	14150	40850 (74%)
	Событие 3	55000	55000	0 (0%)
Последовательность 10 (число шагов: 6)	Событие 1	396000	99400	296600 (75%)
	Событие 2	396000	99400	296600 (75%)
	Событие 3	396000	396000	0 (0%)
Сеть 2				
Атакующий 1				
Последовательность 1 (число шагов: 5)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 2 (число шагов: 6)	Событие 1	283800	46200	237600 (84%)
	Событие 2	283800	244200	39600 (14%)
	Событие 3	283800	283800	0 (0%)
Последовательность 3 (число шагов: 5)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	92400	343200 (79%)
	Событие 3	435600	435600	0 (0%)

Продолжение таблицы П.1

Последова-тельность атаки	**Событие**	**Потери до, €**	**Потери после, €**	**Индекс *AL*, €**
Сеть 2				
Атакующий 1				
Последовательность 4 (число шагов: 6)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	139000	296600 (68%)
	Событие 3	435600	435600	0 (0%)
Последовательность 5 (число шагов: 7)	Событие 1	94600	25150	69450 (73%)
	Событие 2	94600	53750	40850 (43%)
	Событие 3	94600	94600	0 (0%)
Последовательность 6 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 7 (число шагов: 7)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)
Последовательность 8 (число шагов: 7)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	264000	171000 (39%)
	Событие 3	435600	435600	0 (0%)
Последовательность 9 (число шагов: 7)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)
Последовательность 10 (число шагов: 7)	Событие 1	94600	25150	69450 (73%)
	Событие 2	94600	53750	40850 (43%)
	Событие 3	94600	94600	0 (0%)
Атакующий 2				
Последовательность 1 (число шагов: 6)	Событие 1	283800	46200	237600 (84%)
	Событие 2	283800	244200	39600 (14%)
	Событие 3	283800	283800	0 (0%)
Последовательность 2 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 3 (число шагов: 7)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)
Последовательность 4 (число шагов: 7)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	264000	171600 (39%)
	Событие 3	435600	435600	0 (100%)
Последовательность 5 (число шагов: 6)	Событие 1	415800	46200	369600 (89%)
	Событие 2	415800	244200	171600 (41%)
	Событие 3	415800	415800	0 (0%)
Последовательность 6 (число шагов: 5)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 7 (число шагов: 5)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	92400	343200 (79%)
	Событие 3	435600	435600	0 (0%)
Последовательность 8 (число шагов: 5)	Событие 1	94600	25150	69450 (73%)
	Событие 2	94600	53750	40850 (43%)
	Событие 3	94600	94600	0 (0%)
Последовательность 9 (число шагов: 6)	Событие 1	283800	46200	237600 (84%)
	Событие 2	283800	244200	39600 (14%)
	Событие 3	283800	283800	0 (0%)
Последовательность 10 (число шагов: 7)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	264000	171600 (39%)
	Событие 3	435600	435600	0 (100%)
Атакующий 3				
Последовательность 1 (число шагов: 5)	Событие 1	41525	25150	16375 (39%)
	Событие 2	41525	39600	1925 (5%)
	Событие 3	41525	41525	0 (0%)

Продолжение таблицы П.1

Последова-тельность атаки	**Событие**	**Потери до, €**	**Потери после, €**	**Индекс *AL*, €**
Сеть 2				
Атакующий 3				
Последовательность 2 (число шагов: 5)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 3 (число шагов: 6)	Событие 1	415800	119200	296600 (71%)
	Событие 2	415800	119200	296600 (71%)
	Событие 3	415800	415800	0 (0%)
Последовательность 4 (число шагов: 7)	Событие 1	415800	46200	369600 (89%)
	Событие 2	415800	244200	171600 (41%)
	Событие 3	415800	415800	0 (0%)
Последовательность 5 (число шагов: 6)	Событие 1	283800	46200	237600 (84%)
	Событие 2	283800	244200	39600 (14%)
	Событие 3	283800	283800	0 (0%)
Последовательность 6 (число шагов: 5)	Событие 1	67100	25150	41950 (63%)
	Событие 2	67100	39600	27500 (41%)
	Событие 3	67100	67100	0 (0%)
Последовательность 7 (число шагов: 5)	Событие 1	41525	25150	16375 (39%)
	Событие 2	41525	39600	1925 (5%)
	Событие 3	41525	41525	0 (0%)
Последовательность 8 (число шагов: 7)	Событие 1	415800	46200	369600 (89%)
	Событие 2	415800	244200	171600 (41%)
	Событие 3	415800	415800	0 (0%)
Последовательность 9 (число шагов: 6)	Событие 1	283800	46200	237600 (84%)
	Событие 2	283800	244200	39600 (14%)
	Событие 3	283800	283800	0 (0%)
Последовательность 10 (число шагов: 5)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Сеть 3				
Атакующий 1				
Последовательность 1 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 2 (число шагов: 7)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)
Последовательность 3 (число шагов: 7)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	264000	171600 (39%)
	Событие 3	435600	435600	0 (100%)
Последовательность 4 (число шагов: 7)	Событие 1	59400	25150	34250 (58%)
	Событие 2	59400	49500	9900 (17%)
	Событие 3	59400	59400	0 (0%)
Последовательность 5 (число шагов: 7)	Событие 1	49500	25150	24350 (49%)
	Событие 2	49500	49500	0 (0%)
	Событие 3	49500	49500	0 (0%)
Последовательность 6 (число шагов: 7)	Событие 1	52800	25150	27650 (52%)
	Событие 2	52800	49500	3300 (6%)
	Событие 3	52800	52800	0 (0%)
Последовательность 7 (число шагов: 7)	Событие 1	40975	25150	15825 (39%)
	Событие 2	40975	39600	1375 (3%)
	Событие 3	40975	40975	0 (0%)
Последовательность 8 (число шагов: 7)	Событие 1	49500	25150	24350 (49%)
	Событие 2	49500	39600	9900 (20%)
	Событие 3	49500	49500	0 (0%)
Последовательность 9 (число шагов: 6)	Событие 1	42900	25150	17750 (41%)
	Событие 2	42900	39600	3300 (8%)
	Событие 3	42900	42900	0 (0%)

Продолжение таблицы П.1

Последовательность атаки	**Событие**	**Потери до, €**	**Потери после, €**	**Индекс *AL*, €**
Сеть 3				
Атакующий 1				
Последовательность 10 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Атакующий 2				
Последовательность 1 (число шагов: 7)	Событие 1	40975	25150	15825 (39%)
	Событие 2	40975	39600	1375 (3%)
	Событие 3	40975	40975	0 (0%)
Последовательность 2 (число шагов: 7)	Событие 1	49500	25150	24350 (49%)
	Событие 2	49500	39600	9900 (20%)
	Событие 3	49500	49500	0 (0%)
Последовательность 3 (число шагов: 7)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	264000	171600 (39%)
	Событие 3	435600	435600	0 (100%)
Последовательность 4 (число шагов: 7)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)
Последовательность 5 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 6 (число шагов: 7)	Событие 1	42900	25150	17750 (41%)
	Событие 2	42900	39600	3300 (8%)
	Событие 3	42900	42900	0 (0%)
Последовательность 7 (число шагов: 7)	Событие 1	126225	25150	101075 (80%)
	Событие 2	126225	61656,25	64568,75 (51%)
	Событие 3	126225	126225	0 (0%)
Последовательность 8 (число шагов: 7)	Событие 1	663300	25150	638150 (96%)
	Событие 2	663300	195925	467375 (70%)
	Событие 3	663300	663300	0 (0%)
Последовательность 9 (число шагов: 5)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 10 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Атакующий 3				
Последовательность 1 (число шагов: 5)	Событие 1	33000	19800	13200 (40%)
	Событие 2	33000	29700	3300 (10%)
	Событие 3	33000	33000	0 (0%)
Последовательность 2 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 3 (число шагов: 7)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)
Последовательность 4 (число шагов: 7)	Событие 1	435600	25150	410450 (94%)
	Событие 2	435600	264000	171600 (39%)
	Событие 3	435600	435600	0 (100%)
Последовательность 5 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 6 (число шагов: 7)	Событие 1	42900	25150	17750 (41%)
	Событие 2	42900	39600	3300 (8%)
	Событие 3	42900	42900	0 (0%)
Последовательность 7 (число шагов: 7)	Событие 1	49500	25150	24350 (49%)
	Событие 2	49500	39600	9900 (20%)
	Событие 3	49500	49500	0 (0%)

Продолжение таблицы П.1

Последова-тельность атаки	**Событие**	**Потери до, €**	**Потери после, €**	**Индекс *AL*, €**
Сеть 3				
Атакующий 3				
Последовательность 8 (число шагов: 7)	Событие 1	39600	25150	14450 (36%)
	Событие 2	39600	39600	0 (0%)
	Событие 3	39600	39600	0 (0%)
Последовательность 9 (число шагов: 5)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)
Последовательность 10 (число шагов: 7)	Событие 1	303600	25150	278450 (92%)
	Событие 2	303600	264000	39600 (13%)
	Событие 3	303600	303600	0 (0%)

Приложение Р – Копии актов о внедрении

Fraunhofer SIT

Fraunhofer Institute for
Secure Information Technology SIT

Director
Prof. Dr. Michael Waidner

Rheinstrasse 75
64295 Darmstadt

Roland Rieke
Trust and Compliance
Phone + 49 6151 869-284 | Fax -224
roland.rieke@sit.fraunhofer.de
www.sit.fraunhofer.de

Fraunhofer SIT | Rheinstrasse 75 | 64295 Darmstadt

Darmstadt, June 27, 2013

To whom it may concern:

Contribution of Ms. Elena Doinikova to the MASSIF project

As a research coordinator of the MASSIF project, I declare that Ms. Elena Doinikova provided valuable contributions to this project in the following areas:

1. Security metrics for the assessment of the topology, malefactor and attack characteristics;
2. Techniques for the assessment of the computer network security level for static, dynamic and historical data;
3. Architecture of the security assessment module.

All the results are of equal importance to the MASSIF project. The results represent substantial elements of the MASSIF security analysis process.

Research coordinator of MASSIF project

Roland Rieke

Fraunhofer-Institut
Sichere Informationstechnologie SIT
Rheinstr. 75
D-64295 Darmstadt
Postfach 100542
D-64205 Darmstadt

УТВЕРЖДАЮ
Генеральный директор
ГК «Омега»
_____ Я.А.Алейник
«9» января 2017 г.

АКТ

об использовании результатов кандидатской диссертационной работы Дойниковой Елены Владимировны «Оценка защищенности и выбор защитных мер в компьютерных сетях на основе графов атак и зависимостей сервисов»

Настоящий Акт составлен в том, что результаты диссертационной работы Дойниковой Елены Владимировны, а именно:

- методики оценки защищенности компьютерной сети и выбора защитных мер;
- архитектура и программная реализация системы оценки защищенности компьютерной сети и выбора защитных мер

используются ГК «Омега» в рамках рабочего процесса при управлении безопасностью компьютерной сети организации.

Председатель комиссии: _____ Я.А. Алейник

Члены комиссии: _____ О.В. Синицын

_____ О.В. Бутенко

УТВЕРЖДАЮ
Генеральный директор
ООО «Ароматы безопасности»
_____ А.М. Колбанев
«9» января 2017 г.

Акт

об использовании результатов кандидатской диссертационной работы
Дойниковой Елены Владимировны
«Оценка защищенности и выбор защитных мер в компьютерных сетях на основе графов атак и зависимостей сервисов»

Настоящий Акт составлен в том, что результаты диссертационной работы Дойниковой Елены Владимировны были использованы в ООО «Ароматы безопасности» в рамках совместной научно-инновационной деятельности по разработке и внедрению в рамках рабочего процесса системы управления безопасностью компьютерной сети организации, а именно:

1 методика оценки защищенности на основе графов атак и зависимостей сервисов использовалась для получения комплекса показателей, адекватно характеризующих защищенность компьютерной сети;

2 методика выбора защитных мер на основе графов атак и зависимостей сервисов использовалась для поддержки принятия решений по реагированию и позволила ускорить процесс выбора рациональных защитных мер и сократить потери организации из-за остановки бизнес-процессов в результате успешной реализации атакующих действий.

Председатель комиссии: _____ А.М. Колбанев

Члены комиссии: _____ К.А. Шаповалова

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)

«_____»_____ 201 г. № _____

Санкт-Петербург

УТВЕРЖДАЮ
Первый проректор –
проректор по учебной работе
д.т.н., проф. _____ Г.М. Машков
«__»_____ 2017 г.

Акт

об использовании результатов диссертационной работы
Дойниковой Елены Владимировны
«Оценка защищенности и выбор защитных мер в компьютерных сетях на основе графов атак и зависимостей сервисов» в учебном процессе СПбГУТ

Настоящий Акт составлен в том, что результаты диссертационной работы Дойниковой Елены Владимировны, а именно:

1. комплекс показателей защищенности компьютерных сетей на основе графов атак и зависимостей сервисов;
2. методика оценки защищенности компьютерных сетей на основе графов атак и зависимостей сервисов

используются кафедрой Защищенных системы связи (ЗСС) 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 «Программно–аппаратные средства обеспечения информационной безопасности» (рабочая программа дисциплины, регистрационный №02.34.15/1753) и «Основы информационной безопасности» (рабочая программа дисциплины, регистрационный №10.14/94) при чтении курсов лекций, проведении практических и лабораторных работ.

Заведующий кафедрой ЗСС,
кандидат технических наук, доцент А.В. Красов

Ученый секретарь кафедры ЗСС И.А. Ушаков