

Анализ защищенности компьютерных сетей на основе моделирования действий злоумышленников и построения графа атак

И. В. Котенко, М. В. Степашкин

1. Введение

Нарушение информационной безопасности компьютерных сетей может быть вызвано множеством различных причин: наличием уязвимостей в операционных системах (ОС) и приложениях; неверной конфигурацией аппаратного и программного обеспечения (ПО); ошибками, допущенными при настройке контроля доступа; наличием уязвимых или легко атакуемых сервисов и вредоносного программного обеспечения и т. д.

Используя комбинации имеющихся уязвимостей и недостатки в конфигурации сети и применяемой политике безопасности (ПБ), нарушители (как внешние, так и внутренние), в зависимости от своих целей, могут реализовать разнообразные стратегии нападения. Эти стратегии могут быть направлены на различные критические ресурсы сети и включать многошаговые цепочки атакующих действий. В рамках этих цепочек может осуществляться компрометация различных хостов и реализация различных угроз безопасности.

Поэтому при проектировании и эксплуатации компьютерных сетей перед проектировщиком и (или) администратором сети возникает задача проверки того, обеспечивают ли планируемые для применения или уже используемые параметры конфигурации сети, политика безопасности и механизмы защиты необходимый уровень защищенности. Кроме того, на этапе эксплуатации компьютерных сетей довольно часто происходят изменения в ее конфигурации и составе используемого программного и аппаратного обеспечения, поэтому необходимо постоянно производить мониторинг сети, анализ имеющихся уязвимостей и оценку уровня защищенности. На этапе

проектирования основными исходными данными для анализа защищенности выступают спецификации проектируемой сети и политики безопасности, а на этапе эксплуатации - параметры реальной сети.

Возросшая сложность компьютерных сетей и механизмов защиты, увеличение количества уязвимостей и потенциальных ошибок в их использовании, а также возможностей по реализации атак обуславливает необходимость разработки мощных автоматизированных средств (систем) анализа защищенности. Эти системы призваны выполнять задачи по обнаружению и исправлению ошибок в конфигурации сети, выявлению возможных трасс атакующих действий различных категорий нарушителей (по реализации различных угроз безопасности), определению критичных сетевых ресурсов и выбору адекватной угрозам политики безопасности, которая задействует наиболее подходящие в заданных условиях защитные механизмы.

На этапе проектирования могут использоваться различные методы анализа защищенности и определения общего уровня защищенности, например, базирующиеся на основе количественных и качественных методик анализа риска, в том числе на основе математического аппарата теории вероятностей, байесовских сетей, теории возможностей, нечетких множеств и т. п. Перспективным направлением в оценке уровня защищенности являются подходы, основанные на построении представления возможных действий нарушителей в виде деревьев или графов атак и последующей проверки свойств этого дерева (графа) на основе использования различных методов, например, методов верификации на модели (model checking), а также вычисления на базе данного представления разнообразных метрик защищенности.

На этапе эксплуатации компьютерных систем используются пассивные и активные методы анализа уязвимостей. Пассивные методы реализуются на основе анализа журналов регистрации событий, настроек программного и аппаратного обеспечения и т. п. Активные методы сводятся к «тестированию сетей на проникновение», которое выполняется путем реализации различных атакующих действий. Пассивные методы не позволяют оценить возможные трассы проникновения нарушителей, а активные не всегда могут быть применены, так как приводят к нарушению работоспособности отдельных сервисов или системы в целом. Комбинирование пассивного метода (для получения соответствующих данных о текущей конфигурации и реализуемой политике безопасности), процедур построения графов атак и автоматического вывода и проверки (анализа) свойств построенного графа позволяет частично решить две указанные проблемы.

Данная статья посвящена *разработке архитектуры, моделей и системы анализа защищенности (САЗ), базирующейся на формировании графа атак и вычислении разнообразных метрик защищенности*. Предлагаемый подход подразумевает реализацию комплекса следующих функций:

- моделирование действий злоумышленников;
- построение графа возможных атакующих действий, выполняемых из различных точек сети и направленных на реализацию различных угроз безопасности с учетом квалификации нарушителя;
- определение уязвимостей и «узких мест» в защите (наиболее критичных компонентов компьютерной сети);
- вычисление различных метрик защищенности и определение общего уровня защищенности;
- сопоставление полученных метрик с требованиями и выработка рекомендаций по усилению защищенности.

Основное внимание в настоящей статье уделяется анализу защищенности на этапе проектирования компьютерных сетей. Одним из базовых требований к разрабатываемым процедурам анализа защищенности является их чувствительность не только к конфигурации сети, но и к реализуемой политике безопасности.

Работа организована следующим образом. В *разделе 2* представлены краткий обзор близких по тематике работ и основные положения предлагаемого подхода. В *разделе 3* описана обобщенная архитектура предлагаемой системы анализа защищенности. *Раздел 4* содержит краткое описание концептуального представления сценариев атак, используемого для анализа уязвимостей и определения уровня защищенности. В *разделе 5* дается определение основных понятий и процедур, применяемых при формировании графа атак и вычислении метрик защищенности. В *разделе 6* представлена методика определения уровня защищенности, в том числе заданы используемые таксономии и метрики защищенности, правила их расчета и последовательность экспресс-оценки общего уровня защищенности. В *разделе 7* дано описание реализации системы анализа защищенности и примеры ее использования для экспресс-анализа защищенности тестовой компьютерной сети. В *разделе 8* определяется вычислительная сложность реализации подхода и способы ее уменьшения. В *заключении* приведены основные результаты работы и обозначены направления дальнейших исследований.

2. Релевантные работы и сущность предлагаемого подхода

К настоящему времени выполнено множество исследовательских работ, посвященных разработке методов, методик и систем анализа защищенности. Представляемый в настоящем разделе обзор работ не претендует на полноту, но затрагивает основные работы в данной предметной области.

Для оценки возможностей систем, направленных на обеспечение информационной безопасности, должны, прежде всего, использоваться международные и национальные стандарты оценки и управления информационной безопасностью ISO 17799 (BS7799), ISO 15408 и другие, стандарты аудита информационных систем и информационной безопасности CIBIT, SAC, COSO и т. п. В частности, в соответствии с международным стандартом «Общие критерии оценки безопасности информационных технологий» (ISO 15408) оценка безопасности базируется на моделях системы безопасности, состоящих из перечисленных в стандарте функций. В ISO 15408 содержится ряд предопределенных моделей (профилей), описывающих стандартные модули системы безопасности. С их помощью можно не создавать модели распространенных средств защиты самостоятельно, а пользоваться уже готовыми наборами описаний, целей, функций и требований к этим средствам. Простым примером профилей может служить модель межсетевого экрана или СУБД.

В [1–3] излагаются возможные методики анализа рисков для оценки степени защищенности компьютерных систем. В [4] показана взаимосвязь задач анализа защищенности и обнаружения вторжений с задачей управления рисками, даны обзоры основных стандартов в области защиты информации и управления рисками, инструментальных средств для анализа рисков. Работа [5] посвящена процессу анализа защищенности корпоративных автоматизированных систем. Приведен обзор средств анализа защищенности (сетевых сканеров, средств контроля защищенности системного уровня, анализа параметров защиты). Рассмотрена типовая методика анализа защищенности, эффективность которой подтверждена практикой. В [6] раскрываются современные концепции управления рисками, их реализация на практике, инструментальные средства управления рисками. В [7, 8] предложен подход и описана реализация системы управления рисками нарушения информационной безопасности.

В [9] проведен анализ вопросов, стоящих перед исследователями в области метрик безопасности. Авторы утверждают, что существует необ-

ходимость в разработке интегрированной среды для формирования метрик, определяя их цель, значение, единицы измерения, диапазон принимаемых значений и формируя таксономию.

В [10] предложен подход, основанный на понятии сложности по Колмогорову, которая определяет функциональность системы обеспечения безопасности. Так как сложность является фундаментальной характеристикой информации, то данный подход может быть применен без знания детальной спецификации анализируемой системы.

Работы [11, 12] посвящены разработке руководства по метрикам безопасности информационных систем. Данное руководство может быть использовано организациями для оценки адекватности используемых методов и средств защиты информации. Авторами предлагается подход для разработки собственных метрик безопасности. Представлены примеры метрик и их таксономия.

В [13] рассмотрена таксономия метрик безопасности, которая может быть использована исследователями для разработки собственных метрик. Предложенная таксономия основана на представлении авторов о разделении метрик безопасности на следующие классы: (1) объективные / субъективные; (2) качественные/количественные; (3) статические/динамические; (4) абсолютные/относительные; (5) прямые/косвенные. Авторы выделяют две основные группы метрик: организационные и помогающие оценить возможности продукта или системы в области обеспечения безопасности. Первая группа включает, например, метрики, связанные с персоналом, действующим в обеспечении безопасности. Вторая группа включает метрики, связанные с количеством технических объектов и систем (аппаратных или программных), способных выполнять функции по защите информации.

В научных исследованиях используются различные способы представления сценариев атак и построения графов (деревьев) атак для анализа защищенности: деревья атак [14], формальные грамматики [15], раскрашенные сети Петри [16], метод анализа изменения состояний [17], причинно-следственная модель атак [18], описательные модели сети и злоумышленников [19], структурированное описание на базе деревьев [20], использование и создание графов атак для анализа уязвимостей [21], объектно-ориентированное дискретное событийное моделирование [22], модели, основанные на знаниях [23] и т. д.

В [24, 25] предлагается методика анализа графов атак, рассматривается использование метода верификации на модели (model checking), байесовского и вероятностного анализа, описывается генерация событий, возникающих

при реализации атак, исследование их влияния на заданную спецификацию сети и отображение полученных результатов на сценарных графах.

В работе [26] представлен метод оценки уровня защищенности на основе теории игр. В этой работе авторы рассматривают взаимодействие между злоумышленником и администратором как вероятностную игру с двумя игроками и предлагают модель данной игры.

В [27] предложено использовать для анализа уязвимостей компьютерных сетей метод верификации на модели (model checking). В [28–30] представлены алгоритмы для генерации графов сценариев атак, основанные на верификации на модели, использующие символическое и явно выраженное представление состояний. Эти алгоритмы обеспечивают генерацию контр-примеров для определения свойств безопасности и живучести.

В [31] предложен подход к моделированию и анализу различных сценариев атак. Этот подход базируется на использовании высокоуровневого языка спецификации, трансляции спецификаций на этом языке в конструкции системы верификации на модели SPIN и применении методов оптимизации и верификации на модели для автоматического анализа сценариев атак.

В [32] представлен подход, основанный на тестировании на проникновение с использованием формальных моделей компьютерных систем, описана методика расчета метрик безопасности. Авторы случайным образом создают трассы реализации атак, используя множество состояний модели, и оценивают метрики безопасности как функции от полученных трасс.

В [33] рассматривается методика генерации графов атак. В качестве входных данных, система анализа требует базу данных известных атак, конфигурацию сети и информацию об ее топологии, а также профиль злоумышленника. Используя графы, определяются трассы атак с наибольшей вероятностью успеха.

В работе [34] представлены общие метрики, которые могут быть использованы для анализа и проактивного управления нарушениями при функционировании сложных сетей, а также для процедур восстановления их работоспособности.

В [35] предлагается методология и программное средство для анализа уязвимостей компьютерной сети. Средство может автоматически генерировать граф возможных трасс реализации атак, и на основе этого графа могут быть проверены некоторые свойства защищенности.

В [36] рассмотрен подход к оценке анализа рисков по компрометации ресурсов сети на основе графов атак и байесовского анализа.

В [37] предложено использование логического программирования и средств, базирующихся на Datalog, для автоматического анализа уязвимостей компьютерных сетей.

В [38–41] рассмотрен общий подход, способы визуализации и программное средство для топологического анализа уязвимостей компьютерных сетей. Программное средство позволяет построить граф зависимостей между эксплоитами, который задает все возможные пути реализации атак.

Предлагаемый в настоящей статье подход к анализу защищенности компьютерных сетей основывается на тщательном анализе возможных действий нарушителей по реализации различных угроз нарушения безопасности и построении графов этих действий [42–49]. Общий граф атак описывает всевозможные варианты реализации нарушителем атакующих действий. Предполагается, что такой граф строится на основе моделирования действий нарушителя с учетом параметров конфигурации компьютерной сети и правил реализуемой политики безопасности, а также целей, уровня знаний и умений, а также разнообразия местоположения нарушителя, что позволяет исследовать как действия внешних, так и внутренних злоумышленников.

Таким образом, в работе предлагается реализация методики детальной оценки защищенности, основанной на анализе сценариев атак и процессов, происходящих в анализируемой компьютерной сети.

Данный подход позволяет выполнить оценку уровня защищенности компьютерных сетей в условиях, когда нет возможности получить информацию обо всех аспектах их функционирования. Анализ с точки зрения устойчивости к попыткам взлома может также дополнить результаты базового анализа защищенности конкретными примерами, и позволяет сосредоточиться на частных и наиболее важных аспектах работы отдельных приложений.

Основное отличие предлагаемого в подхода от рассмотренных выше подходов заключается в способе построения графа атак (применяется многоуровневое иерархическое представление стратегий действий злоумышленника) и использовании построенного общего графа атак для определения семейства различных показателей (метрик) защищенности, предназначенных для качественного анализа заданной конфигурации сети и реализуемой политики безопасности.

Система анализа защищенности, использующая предложенный подход, предназначена для функционирования на различных этапах жизненного цикла компьютерной сети, включая этапы проектирования и эксплуатации (рис. 1).

На этапе проектирования САЗ оперирует с моделью анализируемой компьютерной сети, которая базируется на спецификациях компьютерной сети и реализуемой политики безопасности. На этапе эксплуатации САЗ взаимодействует с реальной компьютерной сетью.

Таким образом, входными данными для анализа защищенности являются: спецификация планируемой или реализуемой конфигурации информационной системы; спецификация планируемой или реализуемой политики безопасности; уязвимости аппаратного и программного обеспечения; модель нарушителя; множество требований к защищенности информационной системы.

В результате анализа защищенности определяются уязвимости, строятся трассы (графы) возможных атак, выявляются «узкие места» в компьютерной сети, и вычисляются различные метрики безопасности, которые могут быть использованы для оценки общего уровня защищенности компьютерной сети (системы), а также уровня защищенности ее компонентов. Полученные результаты обеспечивают выработку обоснованных рекомендаций по устранению выявленных узких мест и усилению защищенности системы. На основе данных рекомендаций пользователь вносит изменения в конфигурацию реальной сети или в ее модель, а затем, если необходимо, повторяет процесс анализа уязвимостей и оценки уровня защищенности. Таким образом, обеспечивается требуемый уровень защищенности компьютерной сети (системы) на всех этапах ее жизненного цикла.

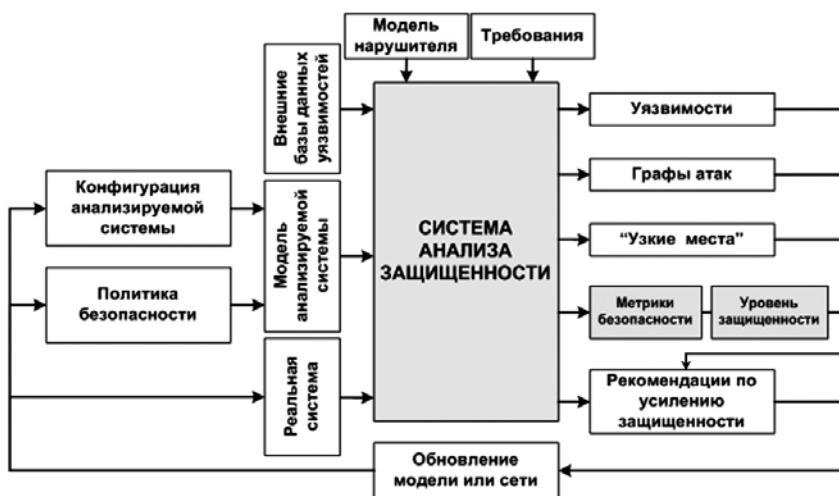


Рис. 1. Обобщенное представление системы анализа защищенности

3. Архитектура системы анализа защищенности

Обобщенная архитектура предлагаемой САЗ, предназначенной для функционирования на этапах проектирования и эксплуатации компьютерных сетей, представлена рис. 2.

САЗ включает следующие элементы: модуль интерфейса пользователя; сетевой интерфейс; модуль формирования внутреннего представления моделей анализируемой сети и политики безопасности; модуль контроля данных; хранилище данных; модуль обновления баз данных (БД) и баз знаний (БЗ); модуль генерации общего графа атак; модуль реализации модели нарушителя; модуль анализа защищенности; модуль генерации отчетов.

На этапе проектирования, САЗ оперирует с моделью анализируемой компьютерной сети, которая базируется на заданных спецификациях анализируемой сети и политики безопасности.

На этапе эксплуатации для построения модели анализируемой сети используется подсистема сбора информации об анализируемой компьютерной сети, состоящая из следующих компонентов (рис. 3): (1) различных источни-

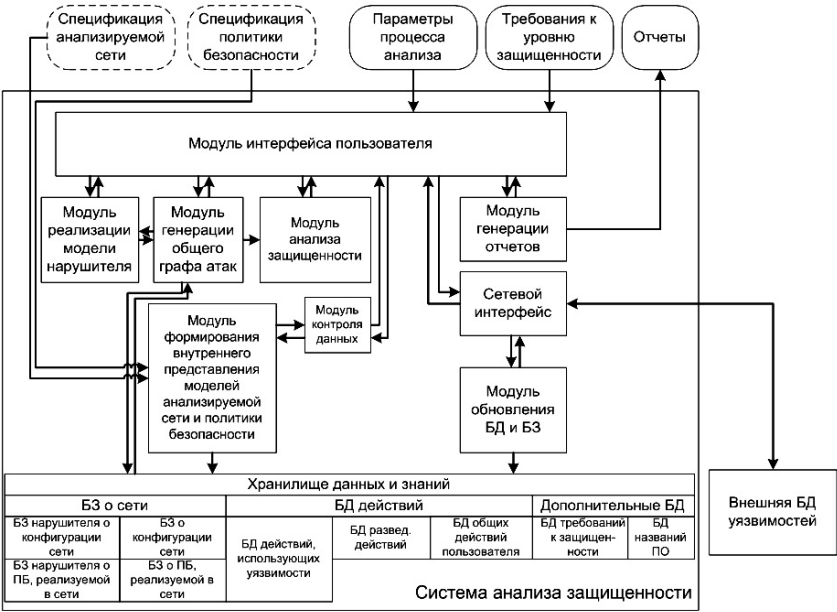


Рис. 2. Обобщенная архитектура САЗ

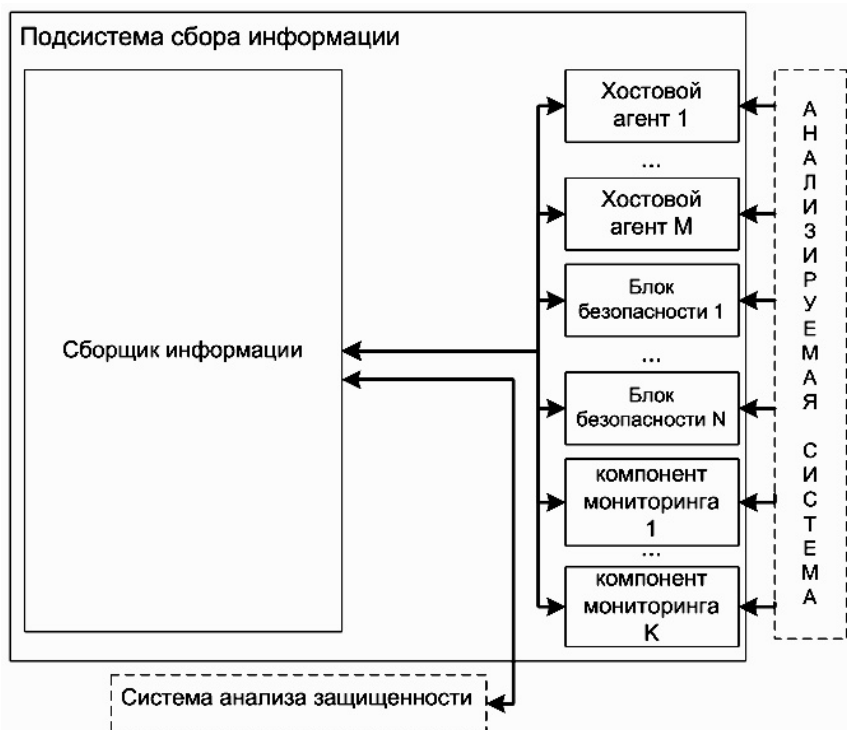


Рис. 3. Архитектура подсистемы сбора информации

ков данных (хостовых программных агентов, блоков безопасности компонентов информационной системы, содержащих параметры безопасности, компонентов проактивного мониторинга безопасности); (2) сборщика информации.

Рассмотрим функции основных модулей предлагаемой САЗ.

Модуль интерфейса пользователя позволяет пользователю (администратору, проектировщику) управлять работой всех компонентов системы, задавать входные данные, просматривать отчеты по анализу защищенности и т. п.

Сетевой интерфейс обеспечивает взаимодействие САЗ с внешней средой: обращение к внешним базам данных уязвимостей за обновлениями; связь с подсистемой сбора информации.

Модуль формирования внутреннего представления анализируемой сети и политики безопасности преобразует данные об анализируемой сети и реализуемой политике безопасности, получаемые от сборщика ин-

формации (на этапе эксплуатации сети) или задаваемые пользователем на языке спецификации конфигурации компьютерной сети и программного и аппаратного обеспечения SDL (System Description Language) и языке спецификации политики безопасности SPL (Security Policy Language) (на этапе проектирования) во внутреннее представление.

Вводимые в САЗ спецификации системы и политики безопасности должны описывать компоненты защищаемой системы (сети) с необходимой степенью детализации — должно быть задано используемое программное обеспечение (в виде названий программных продуктов и их версий).

Если требуемых для анализа уровня защищенности данных недостаточно (например, пользователь не определил версию используемого сетевого сервиса), система должна предложить пользователю ввести необходимую информацию на основе базы данных программного обеспечения.

Для обнаружения некорректных или недоопределенных данных, которые необходимы для анализа защищенности, служит *модуль контроля данных*. Например, пользователь может допустить ошибку в названии сервиса или указать, что на сервере открыт 21 порт, но не определить, какое приложение обрабатывает поступающие на данный порт запросы. Для устранения возникающих при вводе спецификаций модуль контроля обеспечивает пользователю выбор необходимых данных, используя базу названий программного обеспечения.

Хранилище данных состоит из следующих групп баз данных и баз знаний:

- (1) группа баз знаний о сети и реализуемой в ней политике безопасности;
- (2) группа баз данных действий;
- (3) группа дополнительных баз данных.

Термины баз данных и знаний отличаются условно по превалирующему виду представления информации. В первом случае — это фактографическая информация, во втором — информация в виде правил.

Группа баз знаний о сети состоит из четырех баз:

- (1) БЗ о конфигурации анализируемой компьютерной сети (КС);
- (2) БЗ о реализуемой в КС политике безопасности (ПБ);
- (3) БЗ нарушителя о конфигурации анализируемой КС и
- (4) БЗ нарушителя о реализуемой в КС политике безопасности.

Структурно данные БЗ (базы о конфигурации и базы о политике безопасности) попарно совпадают и содержат сведения об архитектуре и

конкретных параметрах компьютерной сети (например, тип и версию ОС, список открытых портов и т. п.) и правилах, описывающих ее функционирование.

Первая БЗ о конфигурации анализируемой КС фактически является внутренним представлением спецификации анализируемой сети, которая используется для формирования результата атакующих действий при построении общего графа атак.

БЗ нарушителя о конфигурации КС является внутренним представлением спецификации анализируемой сети так, как ее представляет себе нарушитель, т. е. как результат реализации последовательности атакующих действий.

БЗ о реализуемой политике безопасности содержит общие правила функционирования сети, например, «локальный пользователь хоста *h* не может запускать приложение *A*». На основе информации из БЗ нарушителя о реализуемой в сети политике безопасности становится возможным планирование последовательности выполняемых нарушителем действий (например, согласно политике безопасности, доступ к файлу *F* разрешен только локальным администраторам, поэтому для чтения данного файла нарушителю необходимо получить требуемые права, т. е. реализовать определенную последовательность действий).

Группа баз данных действий состоит из следующих баз:

- (1) БД действий, использующих уязвимости;
- (2) БД разведывательных действий;
- (3) БД общих действий пользователя.

БД действий, использующих уязвимости (в отличие от других баз данной группы) строится на основе внешней базы данных уязвимостей. Атакующие действия в данной базе делятся на следующие группы:

- (1) действия по получению прав локального пользователя;
- (2) действия по получению прав администратора;
- (3) действия, направленные на нарушение конфиденциальности;
- (4) действия, направленные на нарушение целостности;
- (5) действия, направленные на нарушение доступности.

Примерами действий, содержащихся в данной базе, являются используемые ниже в тестовом примере действия «ServU-local-priv-esc», «Utilman» и др.

БД разведывательных действий содержит действия, направленные на удаленное получение информации о хосте или сети. Описание разведыва-

тельных действий не содержится во внешних базах уязвимостей. Информацию о методах и средствах реализации нарушителем разведывательных действий можно получить лишь экспертным путем. Примерами действий, входящих в данную базу, являются следующие действия из тестового примера: (1) «Nmap-OS», (2) »Ping» и т. д.

База данных общих действий пользователя содержит информацию о возможных действиях пользователя, выполняемых в соответствии с имеющимися у него полномочиями. К таким действиям могут относиться, например, подготовительные действия для выполнения атакующих действий, а также такие действия, как «чтение файла», «копирование файла», «удаление файла», «удаление каталога» и т. п., которые возможно использовать для реализации угроз на нарушение конфиденциальности, целостности и доступности объектов.

Для каждого атакующего действия в БД хранится условие успешной реализации данного действия (например, версия уязвимого программного обеспечения) и результат его воздействия на объект атаки (например, аварийное прекращение работы сетевого сервиса).

Группа дополнительных БД состоит из следующих баз:

- (1) БД требований к защищенности и
- (2) БД названий ПО.

БД требований к защищенности содержит предопределенные экспертным способом наборы значений метрик защищенности, каждый из которых соответствует требованиям к системам определенного класса защищенности, регламентируемым международными стандартами и другими нормативными документами.

База данных названий ПО используется модулем контроля данных для выявления ошибок в используемой спецификации компьютерной сети и формирования рекомендуемых для использования программных средств, в случае отсутствия в спецификации необходимых для анализа защищенности данных.

Модуль обновления БД и БЗ скачивает открытые базы данных уязвимостей (например, NVD [50] или OSVDB [51]) и транслирует их в базу данных атакующих действий.

Модуль генерации общего графа атак производит построение графа атак, моделируя возможные действия нарушителя в анализируемой компьютерной сети, используя информацию о доступных действиях различных типов (атакующих, разведывательных, общих), конфигурации сети и

используемой политике безопасности. Во время формирования графа атак данный модуль расставляет в вершинах метрики защищенности элементарных объектов, на базе которых модуль анализа общего графа атак рассчитывает метрики составных объектов.

Модуль реализации модели нарушителя обеспечивает определение первоначального положения нарушителя, уровня знаний и умений, первичные знания об анализируемой компьютерной сети. Уровень знаний и умений определяет используемый нарушителем набор действий.

Модуль анализа защищенности формирует множество составных объектов общего графа атак (трасс, угроз), производит расчет метрик защищенности, относящиеся к данным объектам, производит оценку общего уровня защищенности компьютерной сети, сравнивает полученные результаты с требованиями, определенными пользователем (если требования были заданы), выявляет слабые места в безопасности и формирует рекомендации по повышению общего уровня защищенности компьютерных сетей.

Модуль генерации отчетов отображает пользователю информацию об обнаруженных уязвимостях в используемом программном и аппаратном обеспечении, слабые места, рекомендации по повышению уровня защищенности компьютерных сетей и т. п.

4. Концептуальное представление сценариев компьютерных атак

Для анализа защищенности компьютерных сетей разработана модель сценариев компьютерных атак, концептуальное представление которой приведено на рис. 4.

Концептуальная модель сценариев атак имеет вид иерархической структуры, состоящей из трех уровней: интегрированного, сценарного и нижнего (уровня действий).

Комплексный уровень определяет множество высокоуровневых целей процесса анализа защищенности, направленных на реализацию основных угроз безопасности (конфиденциальности, целостности, доступности), и множество анализируемых (атакуемых) объектов.

На комплексном уровне может быть обеспечено согласование нескольких сценариев, которые реализуются как одним нарушителем, так и группой нарушителей.

Например, два злоумышленника вступают в сговор для реализации атаки, состоящей из двух этапов: (1) разведка (объекты атаки неизвестны) и (2) реализация угрозы отказа в обслуживании (с указанием множества объектов атаки). Каждый нарушитель выполняет действия одного из этих этапов. Реакцией атакуемой компьютерной сети на проведение первым злоумышленником этапа разведки может быть изменение правил фильтрации сетевого трафика таким образом, чтобы пакеты с хоста первого злоумышленника отбрасывались на граничном хосте сети и не попадали в локальную компьютерную сеть или демилитаризованную зону (ДМЗ). Тогда первый злоумышленник сообщает полученную на этапе разведки информацию второму, который беспрепятственно может реализовать некоторую угрозу, например угрозу отказа в обслуживании.

Сценарный уровень модели компьютерных атак учитывает первичные знания злоумышленника об атакуемой компьютерной сети, его общий уровень знаний и умений, определяет конкретный атакуемый объект (один хост) и цель атаки (например, «определение ОС хоста», «реализация атаки отказа в обслуживании» и т. п.).

Сценарный уровень содержит этапы сценария, множество которых состоит из следующих элементов:

- (1) разведка,
- (2) внедрение (первоначальный доступ к хосту),
- (3) повышение привилегий;
- (4) реализация угрозы;
- (5) сокрытие следов;
- (6) создание потайных ходов.

Нижележащие элементы сценарного уровня служат для детализации цели, достигаемой реализацией сценария.

Нижний уровень (уровень действий) описывает низкоуровневые атакующие действия злоумышленника и используемые эксплойты.

5. Формирование общего графа атак

Алгоритм формирования общего графа атак предназначен для создания графа атак, описывающего всевозможные варианты реализации атакующих действий нарушителем с учетом его первоначального положения, уровня знаний и умений, первоначальной конфигурации компьютерной сети и реализуемой в ней политики безопасности.

На основе общего графа атак производится анализ защищенности информационной системы, определение «узких» мест, формируются рекомендации по устранению обнаруженных уязвимостей с учетом их уровня критичности.

5.1. Объекты графа атак

Все объекты графа атак можно подразделить на базовые (элементарные) объекты и составные.

Вершины графа задаются с использованием базовых объектов. Для формирования различных последовательностей действий нарушителя базовые объекты связываются на графе атак с помощью дуг.

Составные (комбинированные) объекты графа строятся на основе объединения базовых объектов с помощью дуг.

К *базовым объектам* общего графа атак относятся объекты, принадлежащие к типам «хост» и «атакующее действие».

Множество объектов «хосты» включает все обнаруженные нарушителем и атакуемые им сетевые компьютеры (хосты).

Множество объектов «атакующие действия» состоит из всех различных элементарных действий нарушителя.

Атакующие действия разделены на следующие классы:

- действия по получению информации о сети (хосте), т. е. разведывательные действия;
- подготовительные действия (в рамках уже имеющихся у нарушителя полномочий), служащие для создания условий реализации атакующих действий последующих классов;
- действия, направленные на нарушение конфиденциальности;
- действия, направленные на нарушение целостности;
- действия, направленные на нарушение доступности;
- действия, приводящие к получению нарушителем прав локального пользователя;
- действия, приводящие к получению нарушителем прав администратора.

Все атакующие действия можно разделить также на две группы:

- (1) действия, использующие различные уязвимости программного и аппаратного обеспечения, например, «NTP_LINUX_ROOT» (использует уязвимость в сервисе NTP ОС Linux и позволяет нарушителю получить права администратора на атакуемом хосте);

- (2) обычные действия легитимного пользователя системы (в том числе действия по использованию утилит получения информации о хосте или сети), такие как «удаление файла», «остановка сервиса ОС», «использование утилиты ring» и т. п.

Примерами действий по получению информации о сети (хосте) являются:

- (1) «nmap OS» — реализация данного низкоуровневого действия позволяет нарушителю узнать тип и (возможно) точную версию операционной системы;
- (2) «nmap services» — реализация данного низкоуровневого действия позволяет нарушителю получить список открытых на хосте портов;
- (3) «banners» — реализация данного низкоуровневого действия позволяет нарушителю получить названия и версии функционирующих на хосте сетевых сервисов путем анализа баннеров.

Примерами подготовительных действий (в рамках уже имеющихся у нарушителя полномочий), служащих для создания условий реализации атакующих действий других классов являются:

- (1) копирование локальным пользователем утилиты «pipeupadmin» (позволяющей локальному пользователю получить права администратора) перед непосредственным ее запуском;
- (2) смена прав доступа на файл или папку и т. п.

Примерами действий, направленных на нарушение конфиденциальности, целостности и доступности являются:

- (1) просмотр документа, доступ к которому ограничен (как пример нарушения конфиденциальности);
- (2) внесение изменений (вплоть до удаления) в файл (как пример нарушения целостности);
- (3) остановка сервиса/хоста, либо использование различных уязвимостей, таких как «CrachIIS» (уязвимость обнаружена в «Internet Information Services» в «Windows NT 4.0») и т. д. (как пример нарушения доступности).

Примерами действий, приводящих к получению нарушителем прав локального пользователя, являются:

- (1) удаленный подбор злоумышленником пароля пользователя, если на атакуемом хосте функционирует Microsoft Terminal Services в режиме выполнения приложений («Application mode»);

- (2) попытка использования доверительных отношений и т. п.

Примерами действий, приводящих к получению нарушителем прав администратора, являются:

- (1) использование удаленным нарушителем уязвимости «Serv-U MDTM» (уязвимость обнаружена в «RhinoSoft Serv-U FTP Server»);
- (2) использование удаленным нарушителем уязвимости «W2K Remote return into libc» (уязвимыми версиями являются «Windows 2000 SP0» и «Windows XP SP0»);
- (3) использование локальным пользователем утилиты «pipeupadmin» (для «Windows 2000») и т. д.

В представленных в данной работе графах используются также следующие дополнительные элементы:

- сетевой порт;
- элемент, описывающий класс атакующего действия, например, DoS (атакующие действия, направленные на нарушение доступности) или ROOT (атакующие действия, приводящие к получению нарушителем прав администратора).

Данные элементы введены для большей наглядности, так как (1) одно и тоже атакующее действие может быть направлено на различные порты (например, «SYN flood») и (2) из названия атакующего действия часто не очевидна цель его реализации.

К *комбинированным (составным) объектам* отнесем объекты типов «трасса», «угроза» и «граф».

Трасса атаки — это совокупность связанных вершин общего графа атак (хостов и атакующих действий), первая из которых представляет хост, соответствующий первоначальному положению нарушителя, а последняя — не имеет исходящих дуг.

Под угрозой безопасности информации (компьютерной системы) понимается потенциально возможное воздействие на информацию (компьютерную систему), которое прямо или косвенно может нанести урон пользователям или владельцам информации (компьютерной системы).

Согласно данному определению угрозой является любое атакующее действие. Однако, с учетом того, что нарушитель может различными путями достичь такого состояния анализируемой компьютерной сети, которое позволило бы реализовать атакующее действие, необходимо уточнить данное определение.

Под *угрозой* будем понимать множество различных трасс атак, имеющих одинаковые начальную и конечную вершины.

Например, угроза, заключающаяся в реализации удаленным нарушителем состояния отказа в обслуживании Web-сервиса, может быть осуществлена с помощью двух трасс, описанных ниже.

Первая трасса включает следующие действия нарушителя:

- (1) сканирование сети (обнаружение хоста Web_server — действие «PingHosts»);
- (2) сканирование портов хоста Web_server (обнаружение открытого 80-го порта — действие «Nmap-serv»);
- (3) чтение баннера на 80-м порту (тип и версия web-сервиса — IIS 4.0 — действие «Banner»);
- (4) реализация действия «CrashIIS», приводящего к отказу в обслуживании Web-сервиса.

Вторая трасса состоит из следующих действий нарушителя:

- (1) сканирование хоста Firewall_1 (порты, ОС);
- (2) получение прав администратора на хосте Firewall_1 с использованием атакующего действия «Ntp remote buffer overflow»;
- (3) использование отношений доверия — получение прав администратора на хосте «Web_server»;
- (4) используя полученные полномочия, нарушитель останавливает web-сервис, что приводит к требуемому результату.

Разделение атакующих действий по заданным выше классам, позволяет *классифицировать угрозы* следующим образом:

1. Основные угрозы:

- угрозы нарушения конфиденциальности;
- угрозы нарушения целостности;
- угрозы нарушения доступности;

2. Дополнительные угрозы:

- угрозы получения информации о сети (хосте);
- угрозы получения нарушителем прав локального пользователя;
- угрозы получения нарушителем прав администратора.

В общем случае, при успешной реализации нарушителем разведывательных действий, не происходит нарушения конфиденциальности, целостности и доступности информационных ресурсов. Однако, возможно

нарушение конфиденциальности, например, в том случае, если политикой безопасности установлено, что информация о топологии внутренней сети является закрытой.

При успешном получении нарушителем прав локального пользователя, возможности выполнения действий, направленных на нарушение конфиденциальности, целостности и доступности, или на получение прав администратора увеличиваются, так, например, он может нарушить конфиденциальность, целостность и доступность некоторой совокупности объектов хоста, имея только права пользователя.

При успешном получении прав администратора на хосте нарушитель может полностью нарушить конфиденциальность, целостность, доступность всех объектов данного хоста.

В направлении роста степени вложенности все основные объекты графа можно упорядочить следующим образом (стрелка показывает направление увеличения вложенности объектов): хосты, атакующие действия → трассы атак → угрозы → общий граф атак.

5.2. Общее описание алгоритма формирования графа атак

Алгоритм формирования общего графа атак предназначен для создания графа атак, описывающего всевозможные варианты реализации атакующих действий нарушителем с учетом его первоначального положения, уровня знаний и умений, первоначальной конфигурации компьютерной сети и реализуемой в ней политики безопасности.

На основе общего графа атак производится анализ защищенности компьютерной сети, определение «узких» мест, формируются рекомендации по устранению обнаруженных уязвимостей с учетом их уровня критичности.

Алгоритм формирования общего графа атак основан на реализации следующей последовательности действий:

- (1) реализация действий по перемещению нарушителя с одного хоста на другой;
- (2) реализация разведывательных действий по определению «живых» (функционирующих) хостов;
- (3) реализация сценариев (множества действий) разведки для каждого обнаруженного хоста;

- (4) реализация атакующих действий, использующих уязвимости программного и аппаратного обеспечения и общих действий пользователя.

Считаем, что первоначальное положение нарушителя четко определено.

Перемещение нарушителя с текущего хоста на атакуемый хост осуществляется при получении нарушителем на атакуемом хосте прав локального пользователя или администратора в следующих случаях:

- (1) если существует возможность реализации атакующих действий, использующих уязвимости программного и аппаратного обеспечения, требующих у нарушителя наличия прав локального пользователя на атакуемом хосте. Например, реализация атаки с использованием утилиты «pipeupadmin.exe», позволяющей локальному пользователю ОС Microsoft Windows 2000 получить права администратора;
- (2) если переход на атакуемый хост открывает нарушителю доступ к другому сегменту сети;
- (3) если переход на атакуемый хост позволяет нарушителю использовать отношения доверия. Например, если хост В доверяет хосту А, то, реализовав атакующее действие, использующее уязвимость программного обеспечения, и получив права администратора на хосте А, нарушитель может перейти на хост А и, используя отношения доверия, получить права администратора на хосте В.

Примером *разведывательного действия по определению живых хостов* является действие, эмулирующее работу утилиты «ping».

В программной реализации алгоритма формирования общего графа атак используется следующее множество *разведывательных действий*:

- «nmap OS» — реализация данного низкоуровневого действия позволяет нарушителю узнать тип и (возможно) точную версию операционной системы;
- «nmap services» — реализация данного низкоуровневого действия позволяет нарушителю получить список открытых на хосте портов;
- «banners» — реализация данного низкоуровневого действия позволяет нарушителю получить названия и версии функционирующих на хосте сетевых сервисов путем анализа баннеров.

Таким образом, при формировании общего графа атак сценарии атак будут содержать комбинации данных низкоуровневых действий. Так как некоторые комбинации будут приводить к одинаковому результату, выде-

лим следующие сценарии разведывательных действий нарушителя, приводящие к различным результатам:

- «nmap OS»;
- «nmap services»;
- «nmap services» → «banners»;
- «nmap services» → «banners» → «nmap OS».

Из вышеперечисленных сценариев формируется *множество сценариев разведки*.

После реализации каждого сценария из множества сценариев разведки производится проверка условий выполнения *атакующих действий, использующих уязвимости программного и аппаратного обеспечения и общих действий пользователя*. При успешной реализации атакующих действий данной группы, приводящих к получению нарушителем прав локального пользователя или администратора на атакованном хосте, осуществляется проверка необходимости перехода нарушителя на данный хост.

В случае реализации перехода, вышеописанная последовательность действий повторяется для нового положения нарушителя.

5.3. Процедура формирования общего графа атак

Общая процедура формирования графа атак позволяет задать процесс генерации графа на высоком уровне, не вдаваясь в подробности реализации конкретных этапов. Процедура формирования общего графа атак задается следующим образом:

1. Сформировать входные данные.
 - 1.1. Ввести модель реальной сети — «realNetworkModel».
 - 1.2. Сформировать задание на анализ защищенности — «analysisPurpose» (выбор пользователем любой комбинации из следующей тройки (нарушение доступности, нарушение конфиденциальности, нарушение целостности)).
 - 1.3. Учесть уровень знаний нарушителя.
 - 1.4. Инициализировать общий граф атак — «attackGraph».
2. Сформировать начальную модель сети нарушителя — «malefactorNetworkModel».

- 2.1. Создать и добавить в модель нарушителя хост нарушителя — «malefactorHost» (идентификатор создаваемого хоста совпадает с идентификатором хоста нарушителя в модели реальной сети). В дальнейшем под хостом нарушителя будет подразумеваться любой хост сети, на котором в данный момент находится нарушитель.
- 2.2. Создать и добавить в модель нарушителя сетевого концентратора (идентификатор создаваемого концентратора совпадает с идентификатором этого же устройства в модели реальной сети).
- 2.3. Добавить в модель нарушителя его первоначальные знания об анализируемой сети.
3. Вызвать **процедуру формирования графа для определенного положения нарушителя**. Передаваемыми процедуре параметрами являются: «attackGraph», «analysisPurpose», «realNetworkModel», «malefactorNetworkModel», «malefactorPosition» = «malefactorHost»).
4. Сформировать отчет.
5. Конец.

5.4. Процедура формирования графа атак с учетом положения нарушителя

Процедура формирования графа атак для определенного положения нарушителя служит для формирования подграфа общего графа атак для заданного положения нарушителя с учетом текущей (на момент вызова процедуры) модели компьютерной сети нарушителя. Данная процедура позволяет определить всевозможные действия нарушителя, направленные на различные хосты сети, которые он может совершить, находясь на определенном хосте компьютерной сети и имея определенное представление о конфигурации компьютерной сети и реализуемой в ней политике безопасности.

Входными параметрами являются: общий граф атак — «attackGraph»; задание на анализ защищенности — «analysisPurpose»; модель анализируемой компьютерной сети — «realNetworkModel»; модель анализируемой компьютерной сети нарушителя — «malefactorNetworkModel»; положение нарушителя (согласно модели анализируемой КС нарушителя) — «malefactorPosition».

Результатом выполнения процедуры является обновление общего графа атак — «attackGraph».

Процедура формирования графа атак для определенного положения нарушителя представлена ниже:

1. Определить множество доступных для атаки хостов. Данный шаг фактически представляет собой реализацию атаки «pingHosts», которая возвращает хосты с новыми (если промежуточный хост между атакуемым хостом и хостом нарушителя реализует перенаправление портов) или старыми (если нет промежуточных хостов, реализующих перенаправление портов) конфигурациями стека протоколов TCP/IP и связями с сетевым концентратором нарушителя.
 - 1.1. Выбрать в качестве множества анализируемых хостов всех хостов модели реальной сети.
 - 1.2. Инициализировать результирующее множество хостов.
 - 1.3. Для всех хостов из множества анализируемых хостов выполнить следующие действия:
 - 1.3.1. Если хост соединен со свитчем, идентификатор которого совпадает с идентификатором свитча, к которому подключен хост нарушителя, тогда выполнить следующие действия:
 - 1.3.1.1. Если текущий хост реализует перенаправление портов, тогда выполнить следующие действия:
 - 1.3.1.1.1. Для всех правил в таблице перенаправления портов выполнить следующие действия:
 - 1.3.1.1.1.1. Найти хост в реальной модели сети с IP-адресом, заданным в правиле в поле «forward_to» (данное поле содержит IP-адрес хоста, на который перенаправляются сетевые пакеты).
 - 1.3.1.1.1.2. Создать новый хост, используя идентификатор найденного хоста.
 - 1.3.1.1.1.3. Установить имя нового хоста, используя имя реального хоста.
 - 1.3.1.1.1.4. Установить статус нового хоста («Live» — живой, доступный).
 - 1.3.1.1.1.5. Установить права нарушителя для нового хоста («Nobody» — нет прав).

- 1.3.1.1.1.6. Установить флаг «Данный хост интересен нарушителю».
 - 1.3.1.1.1.7. Создать сетевой интерфейс (в качестве идентификатора интерфейса использовать новый MAC-адрес из множества не используемых в тестовом примере адресов).
 - 1.3.1.1.1.8. Добавить к сетевому интерфейсу конфигурацию стека протоколов TCP/IP, в которой используется только IP-адрес, значение которого совпадает с IP-адресом, заданным в правиле перенаправления портов в поле «destinationIP».
 - 1.3.1.1.1.9. Добавить сетевой интерфейс к новому (созданному) хосту.
 - 1.3.1.1.1.10. Создать связь сетевого интерфейса со свитчем нарушителя.
 - 1.3.1.1.1.11. Добавить новый (созданный) хост в результирующее множество хостов.
- 1.3.1.2. В противном случае (если хост не реализует перенаправление портов), выполнить следующие действия:
- 1.3.1.2.1. Создать новый хост, используя идентификатор реального хоста.
 - 1.3.1.2.2. Установить имя нового хоста, используя имя реального хоста.
 - 1.3.1.2.3. Установить статус нового хоста («Live» — живой, доступный).
 - 1.3.1.2.4. Установить права нарушителя для данного хоста («Nobody» — нет прав).
 - 1.3.1.2.5. Установить флаг «Данный хост интересен нарушителю».
 - 1.3.1.2.6. Создать сетевой интерфейс (в качестве идентификатора необходимо использовать MAC-адрес реального хоста).
 - 1.3.1.2.7. Добавить к сетевому интерфейсу конфигурацию стека протоколов TCP/IP, в которой используется только IP-адрес реального хоста.

- 1.3.1.2.8. Добавить сетевой интерфейс к новому (созданному) хосту.
- 1.3.1.2.9. Создать связь сетевого интерфейса со свитчем нарушителя.
- 1.3.1.2.10. Добавить новый (созданный) хост в результирующее множество хостов.
- 1.4. Обновить модель сети нарушителя, используя результирующее множество хостов.
- 2. Для всех хостов модели сети нарушителя, у которых: (1) установлен флаг «Данный хост интересен нарушителю» и (2) хост подключен к тому же свитчу, что и хост нарушителя, выполнить следующие действия:
 - 2.1. Скопировать (сделать резервную копию) модели КС нарушителя («malefactorNetworkModelCopy»).
 - 2.2. Проверить отношения доверия: если атакуемый хост доверяет хосту нарушителя (задаваемому переменной malefactorPosition), тогда вызвать **процедуру получения нарушителем прав локального пользователя или администратора (указав в качестве параметра права нарушителя на хосте, с которого производится атака)**.
 - 2.3. Для каждого сценария разведки из множества recScenarios выполнить следующие действия:
 - 2.3.1. Выполнить сценарий разведки (получить некоторую информацию об атакуемом хосте).
 - 2.3.2. Выбрать согласно заданию на анализ защищенности классы уязвимостей (всего три класса: нарушение доступности, нарушение целостности, нарушение конфиденциальности), для которых будет осуществляться проверка условий успешной реализации атакующих действий.
 - 2.3.3. Проверить для всех уязвимостей в каждом из выбранных классов условия реализации атакующих действий, использующих данные уязвимости. Добавить успешные атакующие действия в общий граф атак.
 - 2.3.4. Выполнить для каждой уязвимости класса «Получение прав пользователя» следующие действия:
 - 2.3.4.1. Если атакуемый хост удовлетворяет условиям атакующего действия, использующего данную уязвимость, тогда добавить уязвимость в общий граф атак.
 - 2.3.5. Если существует хотя бы одна уязвимость, использование которой приводит к получению нарушителем прав пользователя, то-

гда вызвать **процедуру получения нарушителем прав локального пользователя или администратора с параметром gainedRights = «USER»**.

2.3.6. Выполнить для каждой уязвимости класса «Получение прав администратора» следующие действия:

2.3.6.1. Если атакуемый хост удовлетворяет условиям атакующего действия, использующего данную уязвимость, тогда добавить уязвимость в общий граф атак.

2.3.7. Если существует хотя бы одна уязвимость, использование которой приводит к получению нарушителем прав администратора, тогда вызвать **процедуру получения нарушителем прав локального пользователя или администратора с параметром gainedRights = «ROOT»**.

5.5. Процедура получения нарушителем повышенных прав

Данная процедура реализует все необходимые действия, выполняемые при получении нарушителем прав локального пользователя или администратора на хосте компьютерной сети, а также управляет передвижением нарушителя по сети (во время выполнения процедуры принимается решение о необходимости нарушителю перейти на хост, для которого им были получены права пользователя или администратора). Процедура представляет собой метод объекта «хост».

Входными параметрами являются:

- модель анализируемой компьютерной сети — «realNetworkModel»;
- модель анализируемой компьютерной сети нарушителя — «malefactorNetworkModel»;
- положение нарушителя (согласно модели анализируемой КС нарушителя) — «malefactorPosition»;
- получаемые права — «gainedRights» («USER» или «ROOT»).

Результатом работы процедуры является изменение модели нарушителя — «malefactorNetworkModel».

Процедура получения нарушителем прав локального пользователя или администратора задается так:

1. Получить всю информацию о хосте.

1.1. Получить информацию об операционной системе.

- 1.2. Получить информацию о дополнительных параметрах.
- 1.3. Получить информацию о сетевых интерфейсах:
 - 1.3.1. Если сетевых интерфейсов больше одного, выполнить следующие действия:
 - 1.3.1.1. Найти тот интерфейс, через который нарушитель проник на хост.
 - 1.3.1.2. Выставить у других интерфейсов флаг «Данный интерфейс интересен нарушителю».
 - 1.3.2. Если известного нарушителю IP-адреса хоста нет среди его «реальных» адресов (т. е. нарушитель проник на хост с использованием перенаправления портов), тогда выполнить следующие действия:
 - 1.3.2.1. По реальному IP-адресу атакуемого хоста найти идентификатор свитча в реальной сети («switchID»), к которому подключен данный хост.
 - 1.3.2.2. Найти свитч в модели КС нарушителя с идентификатором, найденным на предыдущем шаге («switchInstance»). Если такого свитча нет, выполнить следующие действия:
 - 1.3.2.2.1. Создать свитч с идентификатором из реальной сети («switchID»), запомнить ссылку на объект «switchInstance».
 - 1.3.2.2.2. Добавить свитч в модель компьютерной сети нарушителя.
 - 1.3.2.3. Удалить связь атакуемого хоста модели КС нарушителя от старого свитча.
 - 1.3.2.4. Создать связь атакуемого хоста модели КС нарушителя к свитчу «switchInstance».
2. Установить права нарушителя на данном хосте («gainedRights»).
3. Сбросить флаг «Данный хост интересен нарушителю».
4. Обновить модель КС нарушителя.
5. Принять решение, необходимо ли нарушителю переходить на данный хост.
6. Если решение о переходе положительно, вызвать процедуру формирования графа атак для определенного положения нарушителя, указав, что новое положение нарушителя — данный хост.

6. Методика оценки уровня защищенности

Методика оценки уровня защищенности включает систему различных метрик защищенности (МЗ) и комплекс правил (формул), используемых для их расчета.

Определение значений отдельных метрик и общая оценка уровня защищенности анализируемой компьютерной сети может производиться различными способами.

Выделим *два подхода к оценке уровня защищенности*:

- (1) экспресс-оценка защищенности на основе качественных методик анализа рисков и
- (2) детальное количественное вычисление уровня защищенности (на основе математического аппарата теории вероятностей, байесовских сетей, теории возможностей, нечетких множеств и т. п.). Данный подход позволяет обеспечить большую точность оценки, но требует и большего количества используемых данных и выполняемых вычислений.

В данной статье рассматривается первый подход — экспресс-оценка защищенности на основе качественных методик анализа рисков.

6.1. Таксономии метрик защищенности

Множество всех МЗ строится на основе сгенерированного общего графа атак.

МЗ могут характеризовать защищенность как базовых, так и составных объектов графа атак.

Проведем классификацию используемых метрик защищенности по трем признакам:

- по разделению объектов общего графа атак на базовые и комбинированные (составные);
- в соответствии с порядком вычислений;
- в соответствии с тем, используются ли метрики для определения общего уровня защищенности анализируемой компьютерной сети.

С учетом описанного выше разделения объектов общего графа атак на базовые и комбинированные (составные), множество всех метрик защищенности можно подразделить на следующие группы:

1. МЗ, формируемые по элементарным объектам:
 - МЗ по хостам;
 - МЗ по атакующим действиям;
2. МЗ, формируемые по комбинированным (составным) объектам:
 - МЗ по трассам атак;
 - МЗ по угрозам;
 - МЗ по общему графу атак.

В соответствии с порядком вычислений все МЗ можно разделить на две группы:

- (1) первичные и
- (2) вторичные.

Первичные МЗ получают непосредственно из общего графа атак, вторичные - рассчитываются с использованием первичных.

Для расчета вторичных метрик защищенности множество метрик, рассчитываемых на основе общего графа атак, необходимо дополнить метриками, рассчитываемыми по заданной конфигурации анализируемой компьютерной сети.

В соответствии с тем, используются ли метрики для определения общего уровня защищенности, выделим основные и вспомогательные метрики.

Основные метрики непосредственно используются для получения качественной оценки уровня защищенности анализируемой компьютерной сети.

Вспомогательные метрики служат для построения «детальной картины», описывающей защищенность сети, требуемой, например, для выявления «узких мест» в защите и выработки рекомендаций по повышению защищенности.

В качестве основных определим следующие метрики:

- критичность хоста h ($Criticality(h)$);
- уровень критичности атакующего действия a ($Severity(a)$);
- размер ущерба, вызванного реализацией атакующего действия с учетом уровня критичности атакуемого хоста ($Mortality(a, h)$);
- размер ущерба при реализации трассы S и угрозы T ($Mortality(S)$ и $Mortality(T)$);

- «сложность в доступе» для атакующего действия a , трассы S и угрозы T ($AccessComplexity(a)$, $AccessComplexity(S)$, $AccessComplexity(T)$);
- степень возможности реализации угрозы T ($Realization(T)$);
- уровень риска угрозы T ($RiskLevel(T)$);
- уровень защищенности анализируемой компьютерной сети $SecurityLevel$.

В табл. 1 представлены примеры используемых метрик защищенности. Основные метрики выделены серым цветом.

Таблица 1

Примеры метрик защищенности

№	Обозначение (формула)	Описание	Гр.
0. МЗ, получаемых на основе параметров конфигурации компьютерной сети			
0.1	N^H	Количество хостов в анализируемой компьютерной сети	П
0.2	N^{FH}	Количество межсетевых экранов в анализируемой компьютерной сети	П
0.3	N^{LH}	Количество хостов, функционирующих под управлением ОС семейства Linux	П
0.4	N^{WH}	Количество хостов, функционирующих под управлением ОС семейства Microsoft Windows	П
0.5	N^{HA}	Количество хостов, на которых используются антивирусные программные средства	П
0.6	N^{HF}	Количество хостов, на которых используются персональные средства фильтрации сетевого трафика	П
0.7	N^{HDS}	Количество хостов, на которых используются хостовые системы обнаружения вторжений	П
1. Метрики защищенности по хостам			
1.1	$Criticality(h)$	Критичность хоста — метрика определяется согласно таблице, приведенной в методике оценки уровня защищенности сети	П

Продолжение таблицы 1

Примеры метрик защищенности

N	Обозначение (формула)	Описание	Гр.
2. Метрики защищенности по атакующим действиям			
2.1	$Severity(a)$	Уровень критичности атакующего действия	В
2.2	$Mortality(a, h)$	Размер ущерба, вызванного реализацией атакующего действия a с учетом уровня критичности хоста h	
2.3	$AccessComplexity(a)$	«Сложность в доступе» атакующего действия a (индекс CVSS)	
2.4	V^{VI}	Индекс CVSS «Базовая оценка» («BaseScore») атакующего действия	П
2.5	V^{VI_c}	Индекс CVSS «Воздействие на конфиденциальность» («Confidentiality Impact») атакующего действия	П
2.6	V^{VI_i}	Индекс CVSS «Воздействие на целостность» («Integrity Impact») атакующего действия	П
2.7	V^{VI_a}	Индекс CVSS «Воздействие на доступность» («Availability Impact») атакующего действия	П
2.8	$V^{V_{ac}}$	Индекс CVSS «сложность в доступе» («Access Complexity») атакующего действия	П
3. Метрики защищенности по трассам атак			
3.1	N_R^{VI}	Количество различных уязвимых хостов в трассе (длина трассы, выражаемая в количестве уязвимых хостов)	П
3.2	$N_R^{VI_U}$	Количество различных хостов в трассе, на которых нарушителем получены права локального пользователя	П
3.3	$N_R^{VI_R}$	Количество различных хостов в трассе, на которых нарушителем получены права администратора	П
3.4	N_R^V	Количество различных атакующих действий в трассе (длина трассы, выражаемая в количестве атакующих действий)	П

Продолжение таблицы 1

Примеры метрик защищенности

№	Обозначение (формула)	Описание	Гр.
3.5	V_R^{diff}	Множество различных атакующих действий в трассе	П
3.6	$V_R^{VI} = \sum_{V_R^{diff}} V^{VI} / N_R^V$	Средняя базовая оценка по всем различным атакующим действиям трассы	В
3.7	$V_R^{VIC} = \sum_{V_R^{diff}} V^{VIC} / N_R^V$	Среднее воздействие на конфиденциальность по всем различным атакующим действиям трассы	В
3.8	$V_R^{VIL} = \sum_{V_R^{diff}} V^{VIL} / N_R^V$	Среднее воздействие на целостность по всем различным атакующим действиям трассы	В
3.9	$V_R^{VIA} = \sum_{V_R^{diff}} V^{VIA} / N_R^V$	Среднее воздействие на доступность по всем различным атакующим действиям трассы	В
3.10	$V_R^{VAC} = \max_{V_R^{diff}} \{V^{VAC}\}$	Наивысшая сложность в доступе, требуемой для реализации всех атакующих действий трассы	В
3.11	$Mortality(S)$	Размер ущерба при реализации трассы	
3.12	$Mortality^{max}(S)$	Максимальный размер ущерба при реализации трассы — формула расчета метрики приведена в методике оценки уровня защищенности сети	В
3.13	$AccessComplexity(S)$	Индекс «сложность в доступе» трассы — формула расчета метрики приведена в методике оценки уровня защищенности сети	В
4. Метрики защищенности по угрозам			
4.1	$N_T^{VHmin} = \min \{N_{R_i}^{VH}\},$ $R_i \in T$	Минимальное количество различных уязвимых хостов, используемых при реализации угрозы	В
4.2	$N_T^{VHmax} = \max \{N_{R_i}^{VH}\},$ $R_i \in T$	Максимальное количество различных уязвимых хостов, используемых при реализации угрозы	В
4.3	$N_T^{VHmin} = \min \{N_{R_i}^{VH}\},$ $R_i \in T$	Минимальное количество различных хостов по всем трассам реализации угрозы, на которых нарушителем получены права локального пользователя	В

Продолжение таблицы 1

Примеры метрик защищенности

№	Обозначение (формула)	Описание	Гр.
4.4	$N_T^{VH_U^{\max}} = \max\{N_{R_i}^{VH_U}\},$ $R_i \in T$	Максимальное количество различных хостов по всем трассам реализации угрозы, на которых нарушителем получены права локального пользователя	В
4.5	$N_T^{VH_R^{\min}} = \min\{N_{R_i}^{VH_R}\},$ $R_i \in T$	Минимальное количество различных хостов по всем трассам реализации угрозы, на которых нарушителем получены права администратора	В
4.6	$N_T^{VH_R^{\max}} = \max\{N_{R_i}^{VH_R}\},$ $R_i \in T$	Максимальное количество различных хостов по всем трассам реализации угрозы, на которых нарушителем получены права администратора	В
4.7	$N_T^{V^{\min}} = \min\{N_{R_i}^V\},$ $R_i \in T$	Минимальное количество различных атакующих действий в трассах реализации угрозы	В
4.8	$N_T^{V^{\max}} = \max\{N_{R_i}^V\},$ $R_i \in T$	Максимальное количество различных атакующих действий в трассах реализации угрозы	В
4.9	$V_T^{VI^{\min}} = \min\{V_{R_i}^{VI}\},$ $R_i \in T$	Минимальная средняя базовая оценка по всем трассам реализации угрозы	В
4.10	$V_T^{VI^{\max}} = \max\{V_{R_i}^{VI}\},$ $R_i \in T$	Максимальная средняя базовая оценка по всем трассам реализации угрозы	В
4.11	$V_T^{Vlc^{\min}} = \min\{V_{R_i}^{Vlc}\},$ $R_i \in T$	Минимальное среднее воздействие на конфиденциальность по всем трассам реализации угрозы	В
4.12	$V_T^{Vlc^{\max}} = \max\{V_{R_i}^{Vlc}\},$ $R_i \in T$	Максимальное среднее воздействие на конфиденциальность по всем трассам реализации угрозы	В
4.13	$V_T^{Vli^{\min}} = \min\{V_{R_i}^{Vli}\},$ $R_i \in T$	Минимальное среднее воздействие на целостность по всем трассам реализации угрозы	В

Продолжение таблицы 1

Примеры метрик защищенности

№	Обозначение (формула)	Описание	Гр.
4.14	$V_T^{VI_I} = \max\{V_{R_i}^{VI_I}\},$ $R_i \in T$	Максимальное среднее воздействие на целостность по всем трассам реализации угрозы	В
4.15	$V_T^{VI_A} = \min\{V_{R_i}^{VI_A}\},$ $R_i \in T$	Минимальное среднее воздействие на доступность по всем трассам реализации угрозы	В
4.16	$V_T^{VI_A} = \max\{V_{R_i}^{VI_A}\},$ $R_i \in T$	Максимальное среднее воздействие на доступность по всем трассам реализации угрозы	В
4.17	N_T^R	Количество различных трасс, приводящих к реализации угрозы	П
4.18	$Mortality(T)$	Размер ущерба при реализации угрозы — формула расчета метрики приведена в методике оценки уровня защищенности сети	В
4.19	$Mortality^{\max}(T)$	Максимальный размер ущерба при реализации угрозы — формула расчета метрики приведена в методике оценки уровня защищенности сети	В
4.20	$AccessComplexity(T)$	Индекс «сложность в доступе» угрозы — формула расчета метрики приведена в методике оценки уровня защищенности сети	В
4.21	$Realization(T)$	Степень возможности реализации угрозы — формула расчета метрики приведена в методике оценки уровня защищенности сети	В
4.22	$RiskLevel(T)$	Уровень риска угрозы — формула расчета метрики приведена в методике оценки уровня защищенности сети	В
5. Метрики защищенности по общему графу атак			
а) по хостам			
5.a.1	N_G^{VH}	Количество различных уязвимых хостов на графе	П
5.a.2	$N_G^{VH_v}$	Количество различных хостов на графе, на которых нарушителем получены права локального пользователя	П

Продолжение таблицы 1

Примеры метрик защищенности

№	Обозначение (формула)	Описание	Гр.
5.a.3	$N_G^{VH_R}$	Количество различных хостов на графе, на которых нарушителем получены права администратора	П
б) по атакующим действиям			
5.b.1	N_G^V	Количество различных атакующих действий на графе	П
5.b.2	V_G^{diff}	Множество различных атакующих действий на графе	П
5.b.3	$V_G^{VI} = \sum_{V_G^{diff}} V^{VI} / N_G^V$	Средняя базовая оценка по всем различным атакующим действиям графа	В
5.b.4	$V_G^{VI_c} = \sum_{V_G^{diff}} V^{VI_c} / N_G^V$	Среднее воздействие на конфиденциальность по всем различным атакующим действиям графа	В
5.b.5	$V_G^{VI_l} = \sum_{V_G^{diff}} V^{VI_l} / N_G^V$	Среднее воздействие на целостность по всем различным атакующим действиям графа	В
5.b.6	$V_G^{VI_A} = \sum_{V_G^{diff}} V^{VI_A} / N_G^V$	Среднее воздействие на доступность по всем различным атакующим действиям графа	В
с) по трассам			
5.c.1	N_G^R	Количество трасс атак на графе	В
5.c.2	$N_G^{R^c}$	Количество трасс атак на графе, приводящих к нарушению конфиденциальности	В
5.c.3	$N_G^{R^l}$	Количество трасс атак на графе, приводящих к нарушению целостности	В
5.c.4	$N_G^{R^A}$	Количество трасс атак на графе, приводящих к нарушению доступности	В
д) по угрозам			
5.d.1	N_G^T	Количество угроз на графе	В
5.d.2	$N_G^{T^c}$	Количество угроз на графе, приводящих к нарушению конфиденциальности	В
5.d.3	$N_G^{T^l}$	Количество угроз на графе, приводящих к нарушению целостности	В

Окончание таблицы 1

Примеры метрик защищенности

№	Обозначение (формула)	Описание	Гр.
5.d.4	$N_G^{T^A}$	Количество угроз на графе, приводящих к нарушению доступности	В
е) комбинированные (интегральные)			
5.e.1	A_G^{RH}	Массив количества трасс, проходящих через каждый хост общего графа атак	В
5.e.2	A_G^{VH}	Массив количества различных уязвимостей, обнаруженных на каждом хосте общего графа атак	В
5.e.3	<i>SecurityLevel</i>	Интегральная метрика «Уровень защищенности анализируемой компьютерной сети». Является основной результирующей метрикой	В

Примечание: столбец «Гр.» (сокращенно от «Группа») служит для указания группы, к которой относится метрика защищенности. В данном столбце буквой «П» обозначены первичные метрики; буквой «В» — вторичные.

На основе метрик защищенности, представленных в табл. 1, может быть получена достаточно полная картина защищенности анализируемой информационной системы. Так, например, метрики 5.e.1 и 5.e.2 позволяют определить узкие места в компьютерной сети (хосты, через которые проходит наибольшее количество трасс атак и на которые администратор должен обратить внимание в первую очередь).

Множество метрик, приведенных в табл. 1, не претендует на полноту и может быть дополнено другими метриками, например, отношение количества различных уязвимых хостов к общему количеству хостов в сети (N_G^{VH} / N^H) служит хорошим показателем для быстрой оценки защищенности сети.

Некоторые основные метрики защищенности (например, $Severity(a)$, $AccessComplexity(a)$) и значительная часть вспомогательных метрик рассчитываются на базе подхода **Common Vulnerability Scoring System** [52].

Используемые в CVSS величины также называются метриками. Для устранения путаницы между метриками, используемыми в представлен-

ной таксономии метрик защищенности, и метриками, используемыми в CVSS, последние будем называть индексами CVSS.

Индексы CVSS разделены на три основные группы [52]:

- (1) базовые (base);
- (2) временные (temporal);
- (3) связанные с окружением (environmental).

Множество базовых (base) индексов отражает фундаментальные свойства уязвимостей и состоит из семи индексов:

- (1) **вектор доступа** (Access Vector) — *Local* (для использования уязвимости необходим локальный доступ) и *Remote* (для использования уязвимости необходим удаленный доступ);
- (2) **сложность доступа** (Access Complexity) — *High* (существуют условия на доступ, например, специфические временные рамки, специфические обстоятельства (специфическая конфигурация сервиса), взаимодействие с атакуемым человеком), *Low* (нет специфических условий на доступ, т. е. использование уязвимости возможно всегда);
- (3) **необходимость аутентификации** (Authentication) — *Required* (аутентификация необходима), *Not Required* (для реализации атаки аутентификация не нужна);
- (4) **воздействие на конфиденциальность** (Confidentiality Impact) — *None* (нет воздействия на конфиденциальность), *Partial* (значительное раскрытие информации), *Complete* (полное раскрытие критичной информации);
- (5) **воздействие на целостность** (Integrity Impact) — аналогично с предыдущим пунктом — *None, Partial, Complete*;
- (6) **воздействие на доступность** (Availability Impact) — аналогично с предыдущим пунктом — *None, Partial, Complete*;
- (7) **коэффициент уклона воздействия** (Impact Bias) — *Normal* (конфиденциальности, целостности и доступности присвоен одинаковый вес), *Confidentiality* (конфиденциальности присваивается больший вес, чем целостности и доступности), *Integrity* (аналогично с Confidentiality), *Availability* (аналогично с Confidentiality).

Множество временных (temporal) индексов отражает зависимые от времени характеристики уязвимостей и состоит из трех индексов:

- (1) **использование уязвимости** (Exploitability) — *Unproven* (пока не существует эксплоита), *Proof of Concept* (концептуально код экс-

плоита может быть реализован); *Functional* (эксплоит существует); *High* (уязвимость используется функционально-законченным, мобильным кодом илиexploит не нужен (атака обычными действиями пользователя));

- (2) **уровень устранения уязвимости** (Remediation Level) — *Official Fix* (доступно средство устранения уязвимости от производителя ПО), *Temporary Fix* (существует временное средство устранения уязвимости от производителя ПО), *Workaround* (доступно средство устранения уязвимости от третьей стороны), *Unavailable* (не существует средства устранения уязвимости или его применение невозможно);
- (3) **уровень доверия к сообщениям об уязвимости** (Report Confidence) — *Unconfirmed* (один, неподтвержденный источник, либо несколько противоречивых сообщений), *Uncorroborated* (множество неофициальных источников, возможно включая несколько независимых компаний или научных организаций), *Confirmed* (поставщик ПП подтвердил наличие проблемы в ПО).

Множество индексов, связанных с используемым окружением (environmental), вычисляется пользователем и состоит из двух индексов:

- (1) **уровень потенциальных побочных убытков** (Collateral Damage Potential) — *None* (потенциальный ущерб отсутствует), *Low* (успешное использование уязвимости может привести к незначительному ущербу), *Medium* (успешное использование уязвимости может привести к значительному ущербу), *High* (успешное использование уязвимости может привести к катастрофическому ущербу);
- (2) **распространенность уязвимых систем** (Target Distribution) — *None* (в рабочем окружении отсутствуют уязвимые системы), *Low* (уязвимые системы существуют в рабочем окружении, их количество находится в интервале 1–15 %), *Medium* (количество уязвимых систем в рабочем окружении находится в интервале 16–49 %), *High* (количество уязвимых систем в рабочем окружении находится в интервале 50–100 %).

Базовые и временные индексы рассчитываются поставщиками ПО и координаторами. Индексы, связанные с используемым окружением, *опционально* рассчитываются пользователями или организациями, использующими данное ПО.

Базовые индексы определяют *критичность* уязвимости (атакующего действия, реализующего данную уязвимость).

Обобщенная оценка *критичности* уязвимости рассчитываются по следующей формуле [52]:

BaseScore = round to 1 digit of 10

- * (case AccessVector of local: 0,7 remote: 1,0)
- * (case AccessComplexity of high: 0,8 low: 1,0)
- * (case Authentication of required: 0,6 not-required: 1,0)
- * (case ConfidentialityImpact of none: 0 partial: 0,7 complete: 1,0)
- * (case ImpactBias of normal: 0,333 CNFDNTLTy: 0,5 INTGRTy: 0,25 AVLBLTy: 0,25)
- + (case IntegrityImpact of none: 0 partial: 0,7 complete: 1,0)
- * (case ImpactBias of normal: 0,333 CNFDNTLTy: 0,25 INTGRTy: 0,5 AVLBLTy: 0,25)
- + (case AvailabilityImpact of none: 0 partial: 0,7 complete: 1,0)
- * case ImpactBias of normal: 0,333 CNFDNTLTy: 0,25 INTGRTy: 0,25 AVLBLTy: 0,5))

Таким образом, обобщенная оценка *критичности* уязвимости вычисляется так:

$$BaseScore = round \left(10 \cdot AV \cdot AC \cdot A \cdot (CI \cdot IB^C + II \cdot IB^I + AI \cdot IB^A) \right),$$

где $round()$ — функция округления до десятых, например, $round(0,85) = 0,9$;

$$AV = \begin{cases} 0.7, & Access\ Vector = Local, \\ 1.0, & Access\ Vector = Remote, \end{cases}$$

где *Access Vector* — индекс CVSS «вектор доступа»;

$$AC = \begin{cases} 0.8, & Access\ Complexity = High, \\ 1.0, & Access\ Complexity = Low, \end{cases}$$

где *Access Complexity* — индекс CVSS «сложность доступа»;

$$A = \begin{cases} 0.6, & \text{Authentication} = \text{Required}, \\ 1.0, & \text{Authentication} = \text{Not} - \text{required}, \end{cases}$$

где *Authentication* — индекс CVSS «необходимость аутентификации»;

$$CI = \begin{cases} 0, & \text{Confidentiality Impact} = \text{None}, \\ 0.7, & \text{Confidentiality Impact} = \text{Partial}, \\ 1.0, & \text{Confidentiality Impact} = \text{Complete}, \end{cases}$$

где *Confidentiality Impact* — индекс CVSS «воздействие на конфиденциальность»;

$$IB^C = \begin{cases} 0.333, & \text{Impact Bias} = \text{Normal}, \\ 0.5, & \text{Impact Bias} = \text{Confidentiality}, \\ 0.25, & \text{Impact Bias} = \text{Integrity}, \\ 0.25, & \text{Impact Bias} = \text{Availability}, \end{cases}$$

где *Impact Bias* — индекс CVSS «коэффициент уклона воздействия»;

$$II = \begin{cases} 0, & \text{Integrity Impact} = \text{None}, \\ 0.7, & \text{Integrity Impact} = \text{Partial}, \\ 1.0, & \text{Integrity Impact} = \text{Complete}, \end{cases}$$

где *Integrity Impact* — индекс CVSS «воздействие на целостность»;

$$IB^I = \begin{cases} 0.333, & \text{Impact Bias} = \text{Normal}, \\ 0.25, & \text{Impact Bias} = \text{Confidentiality}, \\ 0.5, & \text{Impact Bias} = \text{Integrity}, \\ 0.25, & \text{Impact Bias} = \text{Availability}, \end{cases}$$

где *Impact Bias* — индекс CVSS «коэффициент уклона воздействия»;

$$AI = \begin{cases} 0, & \text{Availability Impact} = \text{None}, \\ 0.7, & \text{Availability Impact} = \text{Partial}, \\ 1.0, & \text{Availability Impact} = \text{Complete}, \end{cases}$$

где *Availability Impact* — индекс CVSS «воздействие на доступность»;

$$IB^A = \begin{cases} 0.333, & \text{Impact Bias} = \text{Normal}, \\ 0.25, & \text{Impact Bias} = \text{Confidentiality}, \\ 0.25, & \text{Impact Bias} = \text{Integrity}, \\ 0.5, & \text{Impact Bias} = \text{Availability}, \end{cases}$$

где *Impact Bias* — индекс CVSS «коэффициент уклона воздействия».

В соответствии с обобщенной оценкой критичности [53], все уязвимости в базе данных уязвимостей NVD [50] разделены на три класса согласно значению метрики *Severity*:

$$Severity = \begin{cases} \text{Low}, & \text{BaseScore} \in [0.0, 3.9], \\ \text{Medium}, & \text{BaseScore} \in [4.0, 6.9], \\ \text{High}, & \text{BaseScore} \in [7.0, 10.0]. \end{cases}$$

Временные индексы определяют *актуальность* уязвимости в заданный момент времени.

Обобщенная оценка *актуальности* уязвимости и рассчитываются по следующей формуле [52]:

TemporalScore = round to 1 digit of BaseScore
* (case Exploitability of unproven: 0,85 proof-of-concept: 0,9 functional: 0,95 high: 1,00)
* (case RemediationLevel of official-fix: 0,87 temporary-fix: 0,90 work around: 0,95 unavail: 1,00)
* (case ReportConfidence of unconfirmed: 0,90 uncorroborated: 0,95 confirmed: 1,00)

Таким образом, обобщенная оценка *актуальности* уязвимости вычисляется так:

$$TemporalScore = \text{round} (\text{BaseScore} \cdot E \cdot RL \cdot RC),$$

где функция $\text{round}(\)$ — округление до десятых;

$$E = \begin{cases} 0.85, & \text{Exploitability} = \text{Unproven}, \\ 0.9, & \text{Exploitability} = \text{Proof of concept}, \\ 0.95, & \text{Exploitability} = \text{Functional}, \\ 1.0, & \text{Exploitability} = \text{High}, \end{cases}$$

где *Exploitability* — индекс CVSS «использование уязвимости»;

$$RL = \begin{cases} 0.87, & \text{Remediation Level} = \text{Official fix}, \\ 0.9, & \text{Remediation Level} = \text{Temporary fix}, \\ 0.95, & \text{Remediation Level} = \text{Workaround}, \\ 1.0, & \text{Remediation Level} = \text{Unavailable}, \end{cases}$$

где *Remediation Level* — индекс CVSS «уровень устранения уязвимости»;

$$RC = \begin{cases} 0.9, & \text{Report Confidence} = \text{Unconfirmed}, \\ 0.95, & \text{Report Confidence} = \text{Uncorroborated}, \\ 1.0, & \text{Report Confidence} = \text{Confirmed}, \end{cases}$$

где *Report Confidence* — индекс CVSS «уровень доверия к сообщениям об уязвимости».

Индексы, связанные с рабочим окружением, могут использоваться для расстановки приоритетов при планировании действий по устранению уязвимостей.

Обобщенная оценка *критичности окружения* рассчитывается по следующей формуле [52]:

$$\text{EnvironmentalScore} = \text{round to 1 digit of } (\text{TemporalScore} + (10 - \text{TemporalScore}))$$

* (case CollateralDamagePotential of none: 0 low: 0,1 medium: 0,3 high: 0,5)

* (case TargetDistribution of none: 0 low: 0,25 medium: 0,75 high: 1,00)

Таким образом, обобщенная оценка *критичности окружения* уязвимости вычисляется так:

$$EnvironmentalScore = round(TemporalScore + (10 - TemporalScore) \cdot CDP \cdot TD),$$

где функция $round(\)$ — округление до десятых;

$$CDP = \begin{cases} 0, & \text{Collateral Damage Potential} = \text{None}, \\ 0.1, & \text{Collateral Damage Potential} = \text{Low}, \\ 0.3, & \text{Collateral Damage Potential} = \text{Medium}, \\ 0.5, & \text{Collateral Damage Potential} = \text{High}, \end{cases}$$

где *Collateral Damage Potential* — индекс CVSS «уровень потенциальных побочных убытков»;

$$TD = \begin{cases} 0, & \text{Target Distribution} = \text{None}, \\ 0.25, & \text{Target Distribution} = \text{Low}, \\ 0.75, & \text{Target Distribution} = \text{Medium}, \\ 1.0, & \text{Target Distribution} = \text{High}, \end{cases}$$

где *Target Distribution* — распространенность уязвимых систем.

Использование временных индексов CVSS и индексов, связанных с окружением, дает более точную оценку защищенности анализируемой системы, однако, для упрощения методики анализа защищенности, данные группы индексов не будут использоваться в тестовых примерах.

Сложность использования временных индексов и индексов, связанных с окружением, связана с необходимостью их расчета (с привлечением человека) при смене анализируемой сети или обновлении программного и аппаратного обеспечения, в то время как базовые индексы могут быть получены в автоматическом режиме из внешних баз данных уязвимостей.

Индексы CVSS для атакующих действий, использующих различные уязвимости программного и аппаратного обеспечения, могут быть взяты непосредственно из внешних баз данных уязвимостей. Например, индексы для атакующего действия «SYN flood», могут быть получены из базы NVD [50].

Для атакующих действий, составляющих группу обычных действий пользователя и действий по получению информации о хосте или сети, индексы CVSS рассчитываются экспертами, например, для действия Nmap, позволяющему получить информацию о типе функционирующей на хосте ОС или список открытых портов, экспертным путем определены следующие индексы:

AccessVector = Remote; AccessComplexity = Low;
Authentication = Not – required;
ConfidentialityImpact = Partial, IntegrityImpact = None;
AvailabilityImpact = None;
ImpactBias = Normal

Используя формулу для расчета обобщенного индекса критичности атакующего действия для «Nmap» получаем следующее значение:

$$\begin{aligned} BaseScore(Nmap) &= \\ &= round(10 \cdot 1.0 \cdot 1.0 \cdot 1.0 \cdot (0.7 \cdot 0.333 + 0.0 \cdot 0.333 + 0.0 \cdot 0.333)) = \\ &= 2.3(Low) \end{aligned}$$

где $round()$ — функция округления до десятых.

В данном случае эксперт счел необходимым указать, что атакующее действие Nmap частично нарушает конфиденциальность, так как позволяет нарушителю получить информацию о типе ОС.

6.2. Методика экспресс-оценки общего уровня защищенности

Рассмотрим предлагаемую *методику экспресс-оценки общего уровня защищенности информационной системы*.

Данная методика базируется на использовании оценки серьезности (критичности) атакующего действия $Severity(a)$, рассчитываемой на основе обобщенного уровня критичности атакующего действия CVSS, и

применении некоторых процедур оценки из методики анализа рисков FRAP («Facilitated Risk Analysis Process») [54].

Предложенный подход к получению качественной экспресс-оценки защищенности состоит из следующих этапов:

1. Вычисление метрик защищенности базовых и комбинированных (составных) объектов общего графа атак (*Criticality*, *Severity*, *AccessComplexity*, *Realization*).
2. Получение качественных оценок уровня риска для всех угроз (*RiskLevel*).
3. Оценка уровня защищенности анализируемой информационной системы (*SecurityLevel*) на основе полученных оценок уровней риска всех угроз.

Размер ущерба, вызванного успешной реализацией атакующего действия, находится в зависимости от (1) критичности атакуемого хоста и (2) общего уровня критичности атакующего действия. Данную величину обозначим $Mortality(a, h)$.

Критичность хоста ($Criticality(h)$) определяется проектировщиком (администратором) анализируемой компьютерной сети по своему усмотрению по трехуровневой шкале (High, Medium, Low), исходя из назначения данного хоста и выполняемых им функций.

При назначении уровня критичности хоста проектировщик (администратор) может руководствоваться значениями, представленными в табл. 2.

Значения в табл. 2 могут использоваться для оценки критичности хоста, если основным фактором является обеспечение доступности сете-

Таблица 2

Определение критичности хоста ($Criticality(h)$)

Критичность хоста	Тип хоста
High	DNS сервер, корпоративный маршрутизатор, контроллер домена; серверы и рабочие станции, обрабатывающие критическую информацию
Medium	web-, mail- и ftp-серверы, межсетевые экраны
Low	Персональные рабочие станции

вых ресурсов. В этом случае максимальный уровень критичности установлен для хостов, неверное функционирование (или полное прекращение функционирования) которых приводит к невозможности использования ресурсов сети. Например, выход из строя корпоративного межсетевого экрана приводит к (1) невозможности использования внешними пользователями сетевых ресурсов (серверов), расположенных в демилитаризованной зоне; (2) невозможности использования внутренними пользователями ресурсов глобальной сети Интернет.

Далее в сторону уменьшения уровня критичности идут рабочие сервера, функционирование которых (каждого по отдельности) является очень важной составляющей успешной работы организации.

Минимальным уровнем критичности обладают персональные рабочие станции, нарушения в работе которых незначительно влияют на процессы функционирования организации в целом.

Критичность атакующего действия $Severity(a)$ рассчитывается с использованием обобщенной оценки критичности атакующего действия ($BaseScore(a)$) CVSS следующим образом [52]:

$$Severity(a) = \begin{cases} Low, BaseScore(a) \in [0.0, 3.9] \\ Medium, BaseScore(a) \in [4.0, 6.9] \\ High, BaseScore(a) \in [7.0, 10.0] \end{cases}$$

Размер ущерба $Mortality(a, h)$, вызванного успешной реализацией атакующего действия с учетом уровня критичности атакуемого хоста, рассчитывается табл. 3.

Размер ущерба для хоста h с учетом его критичности, вызванного успешной реализацией угрозы, определяется ее последним атакующим действием:

$$Mortality(T) = Mortality(a_T, h_T),$$

где a_T — последнее атакующее действие в угрозе,

h_T — хост, на который направлено действие a_T .

Таблица 3

Определение размера ущерба ($Mortality(a, h)$)

Критичность хоста $Criticality(h)$	Уровень критичности атакующего действия $Severity(a)$		
	Высокий (High)	Средний (Medium)	Низкий (Low)
Высокий (High)	Высокий (High)	Высокий (High)	Средний (Medium)
Средний (Medium)	Высокий (High)	Средний (Medium)	Низкий (Low)
Низкий (Low)	Средний (Medium)	Низкий (Low)	Низкий (Low)

Размер ущерба $Mortality(T)$ при реализации угрозы T можно охарактеризовать следующим образом:

- Высокий ($High$) — остановка критически важных бизнес-подразделений, которая приводит к существенному ущербу для бизнеса, потере имиджа или неполучению существенной прибыли;
- Средний ($Medium$) — кратковременное прерывание работы критических процессов или систем, которое приводит к ограниченным финансовым потерям в одном бизнес-подразделении;
- Низкий (Low) — перерыв в работе, не вызывающий ощутимых финансовых потерь.

Однако, возможна ситуация, когда нарушителем во время реализации угрозы был нанесен гораздо больший ущерб компьютерной сети, чем рассчитанный по последнему атакующему действию. Для учета данной ситуации необходимо ввести метрики максимального размера ущерба при реализации трассы S и угрозы T , рассчитываемые по следующим формулам:

$$Mortality^{\max}(S) = \max_i (Mortality(a_i, h_i)), i \in [1, N_S], a_i \in S,$$

$$Mortality^{\max}(T) = \max_i (Mortality(S_i)), i \in [1, N_T], S_i \in T,$$

где N_S — длина трассы (количество атакующих действий в трассе);

N_T — количество трасс, реализующих угрозу T .

Для получения качественной оценки уровня риска угрозы необходимо оценить степень возможности ее реализации ($Realization(T)$) и воспользоваться методикой FRAP с использованием полученного ранее размера ущерба при реализации угрозы ($Mortality(T)$).

Для определения степени возможности реализации угрозы T воспользуемся индексом CVSS «сложность доступа» ($AccessComplexity(a)$), где a — атакующее действие) из множества базовых индексов CVSS, задаваемых для каждого атакующего действия в графе атак.

Перечислим возможные значения данного индекса:

- *High* — существуют условия на доступ, например, специфические временные рамки, специфические обстоятельства (специфическая конфигурация сервиса), взаимодействие с атакуемым человеком);
- *Low* — нет специфических условий на доступ, т. е. использование уязвимости возможно всегда.

Тогда индекс «сложность в доступе» для трассы атак S будет вычисляться по следующей формуле:

$$AccessComplexity(S) = \begin{cases} High, \exists k \in [1, N]: AccessComplexity(a_k) = High, \\ Low, \forall k \in [1, N] AccessComplexity(a_k) = Low, \end{cases}$$

$$S = \{a_i\}_{i=1}^N$$

где S — сценарий (трасса) атаки;

N — длина трассы (количество атакующих действий).

Для оценки данного индекса для угрозы (совокупности различных трасс, имеющих одинаковые первую и последнюю вершины) воспользуемся следующей формулой:

$$AccessComplexity(T) = \begin{cases} Low, \exists k \in [1, N_s]: AccessComplexity(S_k) = Low \\ High, \forall k \in [1, N_s] AccessComplexity(S_k) = High \end{cases}$$

где $T = \{S_k\}_{k=1}^{N_s}$ — угроза;

N_s — количество различных трасс, реализующих угрозу T ;

$S_k = \{a_i\}_{i=1}^{N_k}$ — сценарий (трасса) атаки;

N_k — количество атакующих действий в трассе.

Тогда степень возможности реализации угрозы T будет рассчитываться по следующей формуле:

$$Realization(T) = \begin{cases} High, & AccessComplexity(T) = Low \\ Low, & AccessComplexity(T) = High \end{cases}$$

Оценка уровня риска угрозы $RiskLevel(T)$ получается в соответствие с правилом, задаваемым матрицей рисков (табл. 4).

Таблица 4

Матрица оценки уровня риска угрозы

Степень возможности реализации угрозы $Realization(T)$	Уровень серьезности (критичности) угрозы $Severity(T)$		
	Высокий (High)	Средний (Medium)	Низкий (Low)
Высокая (High)	A	B	C
Низкая (Low)	B	C	D

Полученная оценка уровня риска может интерпретироваться следующим образом:

- уровень **A** — связанные с риском действия (например, внедрение новых (дополнительных) компонентов защиты или устранение уязвимостей) должны быть выполнены немедленно и в обязательном порядке;
- уровень **B** — связанные с риском действия должны быть предприняты;
- уровень **C** — требуется мониторинг ситуации (но непосредственных мер по противодействию угрозе принимать, возможно, не надо);
- уровень **D** — никаких действий в данный момент предпринимать не требуется.

Исходя из полученных качественных оценок уровня риска для всех угроз, определим уровень защищенности анализируемой компьютерной сети $SecurityLevel$ следующим образом:

$$SecurityLevel = \begin{cases} Green, \forall i \in [1, N_T] RiskLevel(T_i) = D \\ Yellow, \forall i \in [1, N_T] RiskLevel(T_i) \leq C \\ Orange, \forall i \in [1, N_T] RiskLevel(T_i) \leq B \\ Red, \exists i \in [1, N_T] : RiskLevel(T_i) = A, \quad \text{где } D < C < B < A \end{cases}$$

где N_T — количество всех угроз.

7. Реализация системы анализа защищенности и эксперименты

Для демонстрации предложенного подхода к анализу защищенности компьютерных сетей на этапах проектирования и эксплуатации система анализа защищенности была реализована на языке программирования Java с использованием объектно-ориентированного подхода.

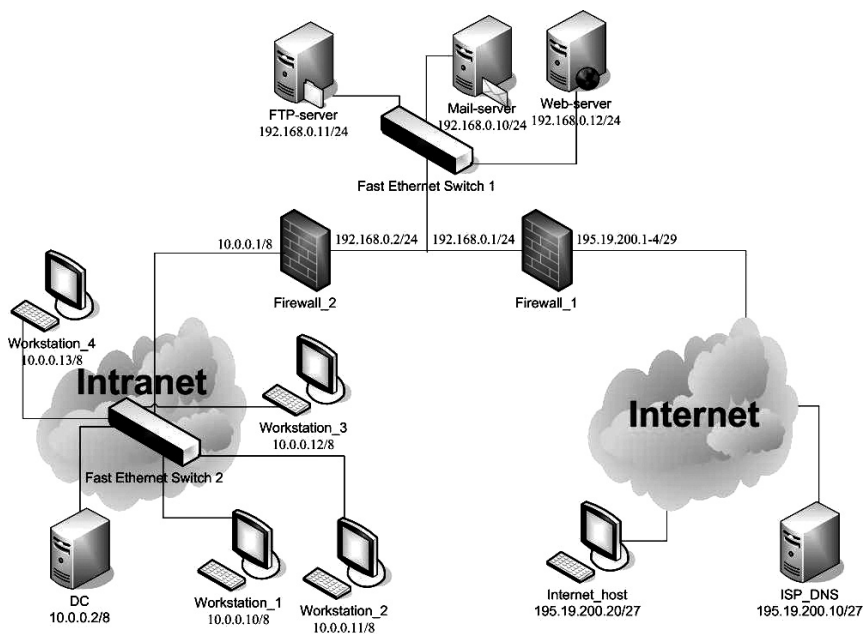


Рис. 5. Структура компьютерной сети

Для проведения экспериментов была специфицирована и реализована тестовая компьютерная сеть.

На рис. 5 представлена структура тестовой компьютерной сети, используемой для проведения ряда экспериментов в задаче анализа защищенности компьютерных сетей на этапах проектирования и эксплуатации.

Тестовая компьютерная сеть состоит из трех подсетей:

- части глобальной сети Internet с IP адресами 195.19.200.*;
- демилитаризованной зоны с IP адресами 192.168.0.*;
- локальной вычислительной сети с IP адресами 10.0.0.*.

7.1. Структура системы анализа защищенности

Структура реализованной САЗ для этапа проектирования представлена на рис. 6.

В реализованной САЗ используются следующие типы действий:

- (1) действия, использующие уязвимости и
- (2) разведывательные действия.

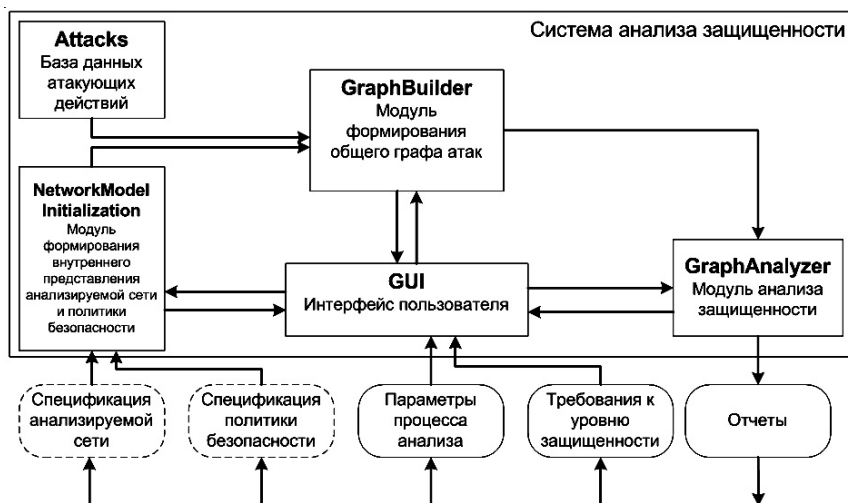


Рис. 6. Структура САЗ

Входными данными для САЗ являются:

- спецификация анализируемой компьютерной сети (на специализированном языке описания системы SDL);
- спецификация политики безопасности, заданная на специализированном языке SPL;
- высокоуровневая цель анализа защищенности: анализ на нарушение конфиденциальности, целостности, доступности.
- параметры, описывающие процесс анализа защищенности (например, параметр «работа с реальной сетью»);
- требования, предъявляемые пользователем к уровню защищенности анализируемой компьютерной сети.

Выходными данными являются отчеты, содержащие список обнаруженных уязвимостей в используемом программном и аппаратном обеспечении, реализуемой политике безопасности, качественную оценку общего уровня защищенности, значения отдельных метрик защищенности, выявленные узкие места в защите, рекомендации по устранению обнаруженных уязвимостей и повышению уровня защищенности анализируемой сети.

Модуль формирования внутреннего представления анализируемой компьютерной сети и политики безопасности преобразует описания компьютерной сети (КС) и политики безопасности (ПБ), представленные с использованием специализированных языков SDL и SPL, во внутреннее представление, с которым работает САЗ.

База данных атакующих действий строится на основе открытых баз уязвимостей. Она содержит условия успешной реализации действий, описание воздействия на атакуемую сеть и множество метрик защищенности по Common Vulnerability Scoring System [52].

Модуль формирования общего графа атак производит построение графа атаки и расставляет в вершинах графа метрики защищенности элементарных объектов, на базе которых модуль анализа общего графа атак рассчитывает метрики составных объектов.

Модуль анализа защищенности производит расчет метрик защищенности и качественную оценку общего уровня защищенности компьютерной сети, сравнивает полученные результаты с требованиями, выявляет слабые места в безопасности и формирует рекомендации по повышению защищенности.

Интерфейс пользователя позволяет пользователю управлять работой всех компонентов системы, задавать входные данные, просматривать отчеты по анализу защищенности.

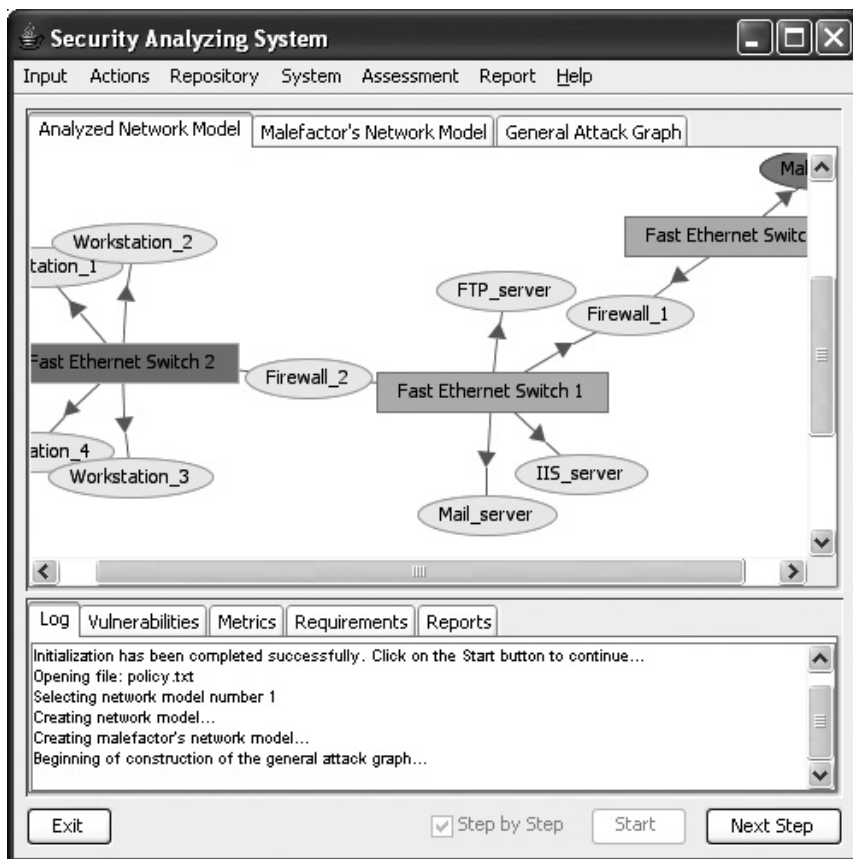


Рис. 7. Интерфейс пользователя CA3

7.2. Интерфейс пользователя

Интерфейс пользователя, реализуемый CA3, представлен на рис. 7. Главное окно CA3 разделено на четыре основные части:

1. Меню, предназначенное для управления работой CA3;
2. Верхняя область рабочей части окна со следующими закладками:
 - «Analyzed Network Model» (модель анализируемой компьютерной сети) — представление модели анализируемой компьютерной сети;

- «Malefactor's Network Model» (модель анализируемой компьютерной сети по представлению нарушителя) - представление модели анализируемой компьютерной сети так, как ее представляет себе нарушитель на данном этапе формирования общего графа атак;
- «General Attack Graph» (общий граф атак) - представление общего графа атак.

3. Нижняя область рабочей части окна со следующими закладками:

- «Log» (журнал регистрации событий) - на вкладке отображается журнал работы САЗ;
- «Vulnerabilities» (уязвимости) - отображает список уязвимых хостов и обнаруженных уязвимостей;
- «Metrics» (метрики защищенности) - отображает метрики защищенности;
- «Requirements» (требования) - вкладка содержит перечень заданных пользователем требований, которым должна удовлетворять анализируемая компьютерная сеть и соответствующие им параметры, полученные при проведении анализа защищенности данной сети;
- «Reports» (отчеты) - на вкладке отображается отчет процесса анализа защищенности (рекомендации по повышению уровня защищенности, действия по устранению обнаруженных уязвимостей и т. п.).

4. Область кнопок управления:

- по нажатию кнопки «Exit» (выход) происходит выход из САЗ;
- флажок «Step by Step» (пошагово) позволяет включить/выключить режим пошагового построения общего графа атак. Данный режим предназначен для наглядной демонстрации процесса построения общего графа атак;
- по нажатию кнопки «Start» (запуск) (активна в момент, когда флажок «Step by Step» сброшен) происходит запуск процедуры построения общего графа атак;
- нажатие кнопки «Next Step» (следующий шаг) (активна в момент, когда флажок «Step by Step» включен) осуществляет реализацию одного атакующего действия и обновление представления нарушителя об анализируемой компьютерной сети и общего графа атак;
- после окончания процесса построения общего графа атак, вместо кнопки «Next Step» появляется кнопка «Analysis», нажатие которой запускает процесс анализа общего графа атак.

7.3. Функционирование системы на этапе проектирования

При работе с САЗ на этапе проектирования, пользователь должен выполнить следующую последовательность действий:

1. Загрузить описание анализируемой компьютерной сети, представленной на специализированном языке SDL, и описание политики безопасности, реализуемой в сети, представленной на специализированном языке SPL;
2. Задать требования к уровню защищенности (не обязательно);
3. Выбрать высокоуровневую цель процесса анализа защищенности;
4. Установить значения параметров, описывающих процесс анализа защищенности компьютерной сети;
5. Произвести анализ защищенности, включающий в себя: (а) построение общего графа атак и (б) его анализ;
6. Выполнить анализ полученных результатов, устранить обнаруженные уязвимости программного и аппаратного обеспечения, а также выявленные слабые места в политике безопасности.

Рассмотрим подробнее каждое действие.

1. *Загрузка описания анализируемой компьютерной сети и реализуемой политики безопасности* осуществляется с помощью пунктов меню «Input → Load Security Policy ...» и «Input → Load System Configuration ...» соответственно (рис. 8). При выборе данных пунктов меню открывается стандартное окно выбора файла. В текущей версии САЗ транслирование SDL и SPL во внутреннее представление не реализовано — в САЗ определены две модели, соответствующие моделям, описанным далее в параграфах «Пример 1» и «Пример 2», выбор между которыми осуществляется посредством указания ее номера (цифры 1 либо 2) в выбираемом для загрузки файле.



Рис. 8. Пункты меню загрузки описания анализируемой компьютерной сети и реализуемой политики безопасности

2. *Задание требований к уровню защищенности анализируемой компьютерной сети* осуществляется с помощью пунктов меню «Assessment → Requirements ...» (рис. 9). Требования задаются путем установки желаемых значений основных метрик защищенности, либо выбора предустановленного шаблона из базы данных требований к уровню защищенности. Если пользователем не заданы требования, то сравнение полученных значений метрик с требуемыми не выполняется, и производится качественный экспресс-анализ уровня защищенности и формирование общего отчета, содержащего список обнаруженных уязвимостей, слабых мест, рекомендации по повышению уровня защищенности и т. д.



Рис. 9. Пункт меню задания требований к уровню защищенности КС

В настоящий момент в САЗ реализована возможность определить требуемый уровень защищенности анализируемой компьютерной сети (рис. 10).



Рис. 10. Определение требуемого уровня защищенности анализируемой сети

3. *Выбор высокоуровневой цели процесса анализа защищенности* осуществляется с использованием пункта меню «Actions → User Goal → Policy Violation» (рис. 11). Пользователю доступны следующие высокоуровневые цели:
- анализ защищенности на предмет нарушения конфиденциальности;
 - анализ защищенности на предмет нарушения целостности;
 - анализ защищенности на предмет нарушения доступности;

- анализ защищенности на предмет нарушения всех вышеперечисленных угроз безопасности.

Исходя из того, что нарушитель, получив права локального пользователя или администратора, может достичь любую из вышеперечисленных целей, то анализ на предмет получения нарушителем прав локального пользователя или администратора производится всегда, независимо от выбранных высокоуровневых целей процесса анализа защищенности.

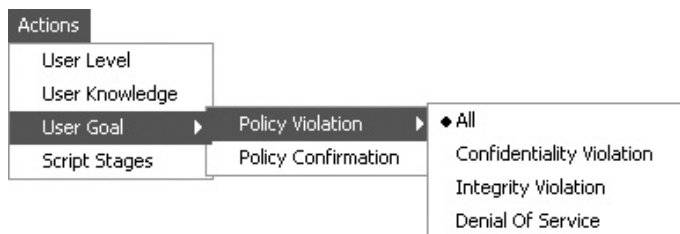


Рис. 11. Меню выбора высокоуровневой цели анализа защищенности



Рис. 12. Пункты меню установки параметров процесса анализа защищенности

4. В качестве *параметра, описывающего процесс анализа защищенности*, используется параметр «работа с реальной сетью» (флажок «Input → Real Network»), (рис. 12). Данный параметр служит для указания того, формируется ли модель анализируемой сети на основе описания (флажок сброшен), или на основе реальных данных, получаемых с помощью программных агентов с действующей сети (флажок выставлен).
5. Запуск *процесса построения общего графа атак* может осуществляться двумя способами:

- стандартным (построение общего графа атак одним кликом);
- в пошаговом режиме (построение общего графа атак в пошаговом режиме).

Результатам процесса построения общего графа атак является граф, отображаемый на закладке «General Attack Graph» и список обнаруженных уязвимостей (закладка «Vulnerabilities»).

Запуск *процесса анализа общего графа атак* осуществляется кнопкой «Analysis», появляющейся вместо кнопки «Next Step» после того, как граф полностью построен (рис. 13). В рамках процесса анализа общего графа производится расчет метрик защищенности, отображаемых на закладке «Metrics».

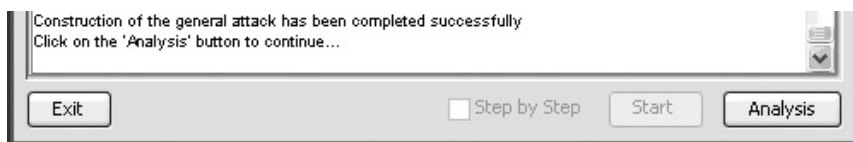


Рис. 13. Окончание процесса построения общего графа атак.

Для запуска процесса его анализа необходимо нажать кнопку «Analysis»

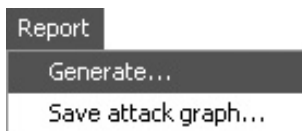


Рис. 14. Пункт меню формирования отчета

Для формирования отчета необходимо выбрать пункт меню «Report → Generate ...» (рис. 14).

Отчет содержит: (1) информацию об обнаруженных уязвимостях, сгруппированных по хостам; (2) метрики безопасности и (3) требования (к общему уровню защищенности анализируемой компьютерной сети) и полученный в процессе анализа уровень защищенности.

Пункт меню «Report → Save attack graph ...» обеспечивает доступ к функции сохранения общего графа атак в файл формата CSV (comma-separated values), который может быть импортирован в «Microsoft Visio» версии ниже 2002 (в Visio 2003 данная функциональность была удалена компанией-разработчиком).

Устранение уязвимостей и слабых мест осуществляется пользователем путем обновления описания анализируемой компьютерной сети и политики безопасности.

Рассмотрим функционирование системы анализа защищенности компьютерных сетей на этапе проектирования на тестовых примерах (Пример 1 и Пример 2).

Первый пример отражает анализ защищенности компьютерной сети, в которой присутствует ряд уязвимостей в используемом программном обеспечении и реализуемой политике безопасности (низкий уровень защищенности).

Второй пример соответствует первому после устранения обнаруженных уязвимостей программного обеспечения и изъянов в политике безопасности.

Раскроем ряд понятий, используемых в описании функционирования САЗ с тестовой компьютерной сетью.

Описание компьютерной сети, заданное на специализированном языке System Description Language (SDL), позволяет определить следующие элементы:

- топологию сети (множество хостов и сетевых концентраторов);
- информацию о функционирующих операционных системах на хостах сети;
- информацию о настройках стека протоколов TCP/IP;
- информацию о сервисах и т. п.

Описание политики безопасности, заданное на специализированном языке Policy Description Language (PDL), позволяет определить следующие элементы:

- правила фильтрации сетевого трафика для граничных хостов;
- отношения доверия.

Правила фильтрации сетевого трафика задаются в виде набора таблиц [55]. В задаче анализа защищенности компьютерных сетей нас будут интересовать правила перенаправления портов («port forwarding»), задаваемые таблицами PREROUTING и FORWARD (считаем, что все входящие соединения, не описанные в таблице, **запрещены**). Данные правила для использования с тестовой сетью можно представить в виде одной таблицы.

Будем считать, что **хост 1 доверяет хосту 2**, если любому пользователю хоста 1, прошедшему процесс аутентификации, при обращении к хосту 2 имеет те же полномочия, что и на хосте 1. Например, пусть хост Firewall_1

доверяет хосту FTP_SERVER и пусть на хосте FTP_SERVER существует пользователь Tigra с правами локального пользователя USER. Тогда, если пользователь Tigra прошел аутентификацию на хосте FTP_SERVER, то при обращении к хосту Firewall_1 он будет также иметь права локального пользователя.

7.4. Пример 1

Входными данными для примера 1 являются:

1. Топология сети, сервисы, ОС и т. п. соответствуют описанию тестовой компьютерной сети;
2. Правила перенаправления портов для хоста Firewall_1 приведены в табл. 5;
3. Firewall_1 и Firewall_2 доверяют всем хостам из ДМЗ;
4. Нарушитель находится во внешней сети на хосте Malefactor и обладает на нем правами администратора;
5. В задании на анализ защищенности указано, что необходимо оценить все типы угроз (на нарушение целостности, доступности, конфиденциальности);
6. В задании на анализ защищенности указано, что необходимо провести анализ всех хостов ДМЗ и ЛВС;
7. Пользователь в качестве требований к анализируемой сети указал, что данная сеть должна иметь уровень защищенности лучше, чем Orange;
8. в текущей версии одним из ограничений САЗ является отсутствие в нем общих действий пользователя.

Таблица 5

Правила перенаправления портов для хоста Firewall_1

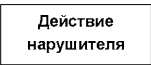
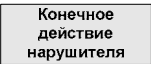


Комментарий	Destination (целевой хост)		Forward to ... (перенаправить на ...)	
	IP	Port	IP	Port
Web_server	195.19.200.3	80	192.168.0.12	80
FTP_server	195.19.200.2	21	192.168.0.11	21
MAIL_server POP3	195.19.200.4	110	192.168.0.10	110
MAIL_server SMTP	195.19.200.4	25	192.168.0.10	25
MAIL_server RDC	195.19.200.4	3389	192.168.0.10	3389

Для примера 1 получается большой общий граф атак. Его представление в виде слепка экрана САЗ (рис. 15) и его последующий анализ значительно затруднены. Поэтому для более качественного представления (с целью более наглядной демонстрации примера) он был построен вручную.

В табл. 6 представлены условные обозначения элементов, используемых на общем графе атак.

Таблица 6

Условные обозначения используемых элементов

 Действие нарушителя	описывает реализацию одного атакующего действия нарушителя
 Конечное действие нарушителя	обозначает конечное действие нарушителя (не имеющее дальнейшего развития)
 Положение нарушителя ПРАВА	служит для обозначения положения нарушителя (хост) и его прав на данном хосте
 Атакуемый хост (критичность)	служит для обозначения атакуемых хостов и их уровня критичности

На графе используются следующие обозначения:

- у хостов в скобках указан их уровень критичности $Criticality(h)$;
- для атакующих действий указан вектор, состоящий из следующих компонентов:
 - ♦ уровня критичности атакующего действия $Severity(a)$;
 - ♦ индекса CVSS «сложность в доступе» атакующего действия $AccessComplexity(a)$.

На рис. 16 представлен общий граф атак, соответствующий вышеописанным входным данным. Кратко рассмотрим построение данного графа атак. В дальнейшем считаем, что действия нарушителя реализует система анализа защищенности.

Нарушитель, находясь на хосте «Malefactor», реализует атаку «Ping Hosts», позволяющую определить живые хосты. Согласно заданному описанию анализируемой компьютерной сети, результатом выполнения данного действия является получение нарушителем информации о четырех хостах (FTP_server, Web_server, Mail_server и Firewall_1) с IP-адресами 195.19.200.1-4 (реально это один хост Firewall_1, однако этого

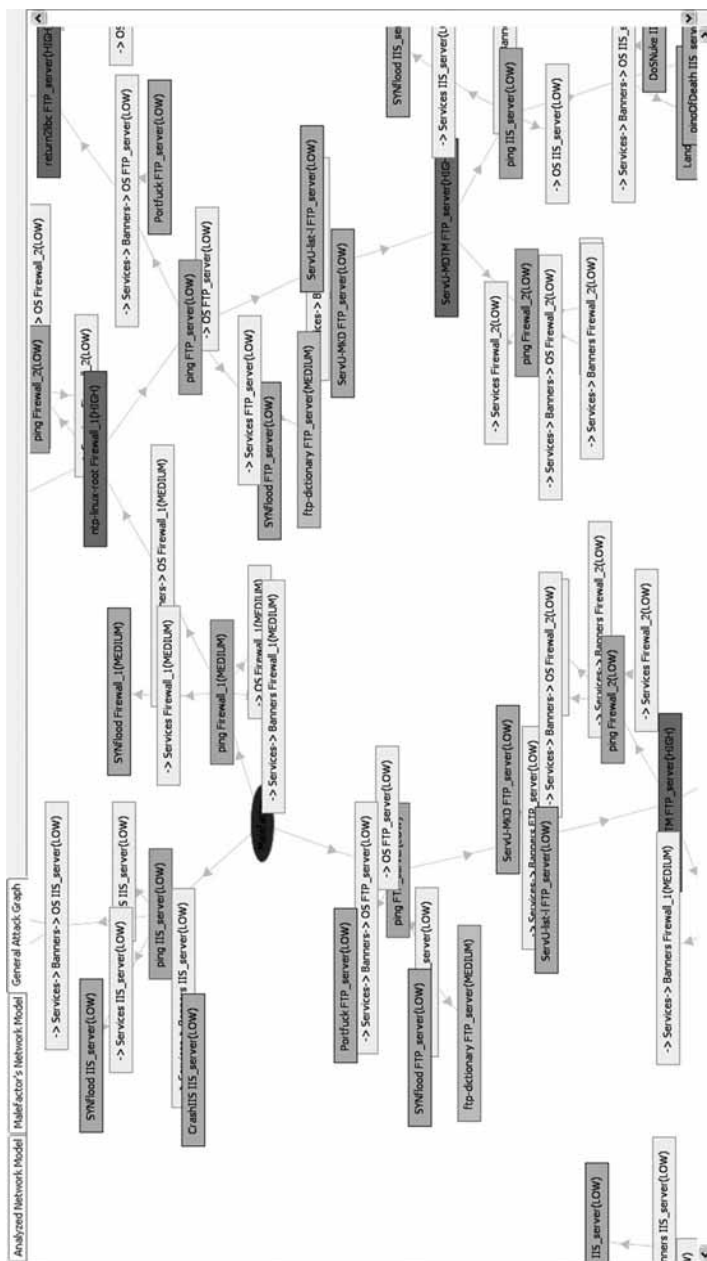


Рис. 15. Общий граф атак для примера 1, построенного с использованием системы анализа защищенности

нарушитель не знает). Далее нарушитель производит анализ каждого хоста отдельно.

Рассмотрим подробнее анализ хоста с IP-адресом 195.19.200.2.

Согласно информации из базы данных разведывательных действий нарушителю доступны четыре сценария разведки:

- (1) «Nmap serv» (определение множества открытых портов);
- (2) «Nmap OS» (определение типа и версии ОС);
- (3) «Nmap serv» + «Banner» (определение множества открытых портов и идентификация сервисов);
- (4) «Nmap serv» + «Banner» + «Nmap OS».

После реализации каждого из сценариев разведки (в представленной последовательности) нарушитель производит проверку, удовлетворяет ли полученная о хосте информация условиям выполнения атакующих действий, использующих уязвимости.

Действия, использующие уязвимости, которые уже были отображены на графе после одного из сценариев разведки, не отображаются повторно после последующих сценариев. Это реализовано для того, чтобы избежать излишнего загромождения графа. Например, атакующее действие «SYN flood» может быть реализовано сразу же после первого сценария разведки («Nmap serv»). Оно может быть реализовано также после третьего и четвертого сценариев. Однако после данных сценариев на графе атак не отображается действие «SYN flood».

Результатом действия «Nmap serv» для хоста с IP-адресом 195.19.200.2 является перечень открытых портов на хосте FTP_server (открытый порт один — 21), так как согласно таблице перенаправления портов входящие соединения на адрес 195.19.200.2 : 21 (где 21 — порт назначения) перенаправляются на 192.168.0.11 : 21. Таким образом, нарушитель определяет наличие одного открытого порта, который может быть атакован с использованием атакующего действия «SYN flood».

Информация, полученная нарушителем после реализации второго сценария разведки («Nmap OS»), не позволяет реализовать какое-либо атакующее действие, использующее уязвимости.

Реализовав третий сценарий разведки, нарушитель может использовать три действий, использующих уязвимости: (1) подбор пароля («FTP dict»); (2) атака на отказ в обслуживании («ServU-MKD»); (3) атака по повышению привилегий («ServU-MDTM»).

Первые два действия являются конечными («FTP dict» является конечным из-за ограничения в текущей версии CA3, а именно из-за отсутст-

вия в нем реализации общих действий пользователя). Выполнив третье действие, нарушитель получает права администратора, что дает ему всю информацию о хосте FTP_server. Получение прав администратора позволяет нарушителю осуществить переход на данный хост для последующей реализации атакующих действий, направленных на другие хосты.

Получив информацию о FTP_server'е, нарушитель обнаруживает, что реальный IP-адрес хоста (192.168.0.11) не совпадает с тем, который «знал» нарушитель (195.19.200.2). Следовательно, в анализируемой сети реализуется перенаправление портов, а значит, нарушитель попал в другую сеть. Этот факт является решающим фактором при решении вопроса об изменении положения нарушителя с текущего хоста (Malefactor) на захваченный (FTP_server).

Изменив положение, нарушитель реализует действие «Ping Hosts» и обнаруживает в сети четыре хоста, которые последовательно анализируются по вышеприведенной схеме. Кроме того, учитываются отношения доверия, задаваемые политикой безопасности, которые позволяют сразу же получить права администратора для хостов Firewall_1 и Firewall_2.

Рассмотрим подробнее решение вопроса о необходимости перехода нарушителя на другие хосты. Данный вопрос решается в двух случаях:

- (1) нарушителем получены права локального пользователя на атакуемом хосте;
- (2) нарушителем получены права администратора на атакуемом хосте.

Для первого случая одним из факторов, влияющих на решение вопроса о переходе, является возможность реализации нарушителем действий по повышению привилегий (получению прав администратора), которые могут быть реализованы только локально.

В общем случае, при решении вопроса о переходе на другой хост необходимо учитывать следующие факторы:

- (1) если хост имеет несколько сетевых интерфейсов, изменение положения нарушителя дает возможность ему проникнуть в другие сети;
- (2) могут быть учтены отношения доверия (получение прав администратора или пользователя на других хостах данной сети).

Дальнейшее построение общего графа атак производится аналогичным образом.

В результате проведенного анализа защищенности компьютерной сети были получены следующие данные:

1. Общее количество хостов в сети: $N^H = 11$;

2. Общее количество различных уязвимых хостов: $N_G^{VH} = 11$;
3. $N_G^{VH} / N^H = 1$;
4. Слабые места в сети (по количеству проходящих через данные вершины трасс атак): Firewall_1, FTP_server, ...
5. Критические уязвимости: NTP_LINUX_ROOT, Serv-U MDTM, ...
6. На графе существуют трассы, а следовательно и угрозы с $Mortality^{\max}(T) = High$ (например, трасса Malefactor-Ping-FTP_server(Nobody)-Nmap serv-Banner-ServU MDTM- FTP_server(Root) ...) и с $Realization(T) = High$.
7. Следовательно $SecurityLevel = Red$. Данный уровень защищенности компьютерной сети не удовлетворяет заданным пользователем требованиям (лучше, чем Orange) и требует немедленных действий по устранению обнаруженных уязвимостей программного обеспечения и слабых мест реализуемой политики безопасности.

Подробное описание представления результатов анализа защищенности в САЗ рассматривается в примере 2.

7.5. Пример 2

Входными данными для примера 2 являются:

1. Топология сети, сервисы, ОС и т. п. соответствуют описанию тестовой компьютерной сети, за исключением:
 - на хосте Firewall_1 уязвимый сервис ntp обновлен на последнюю версию, т. е. известных уязвимостей нет;
 - на хосте Ftp-server уязвимый сервис Serv-u обновлен на последнюю версию;
 - на хостах Ftp-server и Mail-server обновлена ОС;
 - на хосте Web-server обновлена ОС и IIS на последние версии.
2. Правила перенаправления портов для хоста Firewall_1 приведены в табл. 7 (отличие с примером 1 заключается в отключении возможности внешним пользователям использовать сервис Microsoft Remote Desktop Connection).;
3. Firewall_1 и Firewall_2 доверяют всем хостам из ДМЗ.
4. Пользователи ftp-сервиса не имеют учетных записей на сервере.



5. Нарушитель находится во внешней сети на хосте Malefactor и обладает на нем правами администратора.
6. Политикой безопасности установлены правила, делающие невозможным подбор пароля к предоставляемым сетевым сервисам (ftp, pop3),

- т. е. установлены ограничения на количество неверных вводов пароля, после которых учетная запись блокируется.
7. Для межсетевых экранов Firewall_1 и Firewall_2, серверов демилитаризованной зоны администратором установлен уровень критичности Medium.
 8. В задании на анализ защищенности указано, что необходимо оценить все типы угроз (на нарушение целостности, доступности, конфиденциальности).
 9. В задании на анализ защищенности указано, что необходимо провести анализ всех хостов ДМЗ и ЛВС.

Таблица 7

Правила перенаправления портов для хоста Firewall_1

Комментарий	Destination (целевой хост)		Forward to ... (перенаправить на ...)	
	IP	Port	IP	Port
Web_server	195.19.200.3	80	192.168.0.12	80
FTP_server	195.19.200.2	21	192.168.0.11	21
MAIL_server POP3	195.19.200.4	110	192.168.0.10	110
MAIL_server SMTP	195.19.200.4	25	192.168.0.10	25

В табл. 8 представлены условные обозначения элементов, используемых на общем графе атак при построении с использованием САЗ.

На рис. 17 представлен общий граф атак, соответствующий вышеописанным входным данным так, как он представляется в САЗ.

В результате проведенного анализа защищенности компьютерной сети были получены следующие данные (слепки экранов системы анализа защищенности представлены ниже):

1. Общее количество хостов в сети: $N^H = 11$.
2. Общее количество различных уязвимых хостов: $N_G^{VH} = 4$.
3. $N_G^{VH} / N^H = 0.36$.
4. Для всех трасс S $Severity(S) = Low$. Тогда, для всех угроз $Severity(T) = Low$.

5. Для всех угроз степень возможности их реализации $Realization(T) = High$.
6. Следовательно $SecurityLevel = Yellow$. Данный уровень защищенности компьютерной сети не предполагает введение дополнительных средств защиты информации, а требует проведения мониторинга.

На вкладках «Vulnerabilities», «Metrics» и «Reports» используются следующие сокращения:

- М: $Mortality(a, h)$ — уровень критичности атакующего действия a (зависит от уровня критичности хоста h);
- АС: $AccessComplexity(a)$ — индекс CVSS «сложность в доступе» для атакующего действия a .

Список обнаруженных уязвимостей и способы их устранения представлены в окне «Vulnerabilities» (рис. 18). Как видно из рисунка, не существует механизмов защиты от обнаруженных уязвимостей, т. е. у администратора больше нет способов повлиять на текущий уровень защищенности анализируемой компьютерной сети.

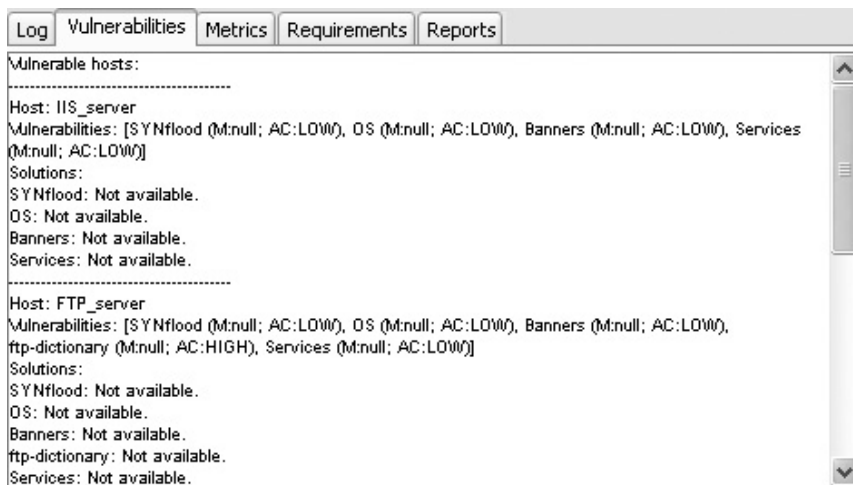


Рис. 18. Окно «Vulnerabilities» для примера 2

Совокупность метрик защищенности, их значения, список угроз и значения общего уровня защищенности представлены в окне «Metrics» (рис. 19).

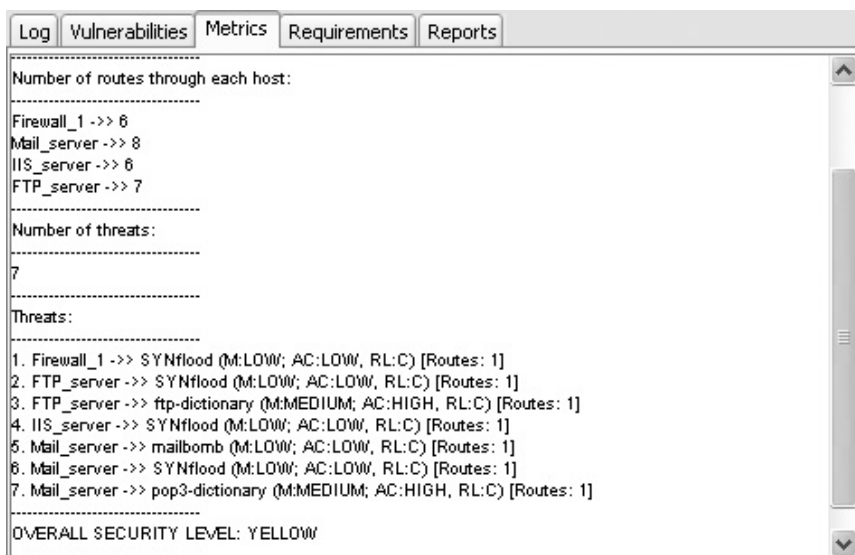


Рис. 19. Метрики защищенности, их значения, список угроз и значение общего уровня защищенности

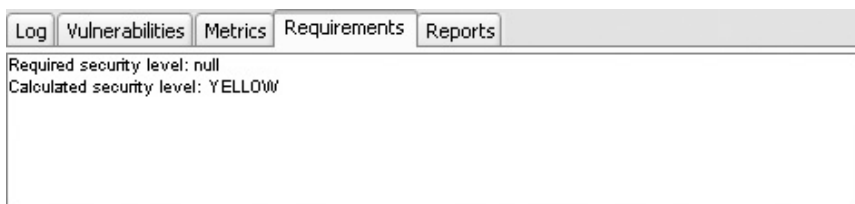


Рис. 20. Окно «Requirements»

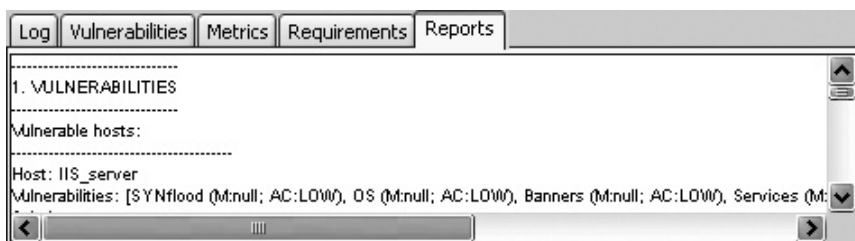


Рис. 21. Окно «Reports»

Отображение требуемого и полученного в результате анализа защищенности уровня защищенности компьютерной сети осуществляется в окне «Requirements» (рис. 20). На рисунке отображена ситуация, когда пользователь не задал требуемый уровень защищенности.

Сформированный с помощью пункта меню «Report → Generate ...» отчет отображается в окне «Reports» (рис. 21).

8. Вычислительная сложность реализации подхода

Определим *сложность алгоритма формирования дерева атак* F как число выполненных нарушителем действий в предположении, что производится анализ *сети без деления на сегменты*.

Введем следующие обозначения:

- (1) \mathbb{H} — множество анализируемых хостов в компьютерной сети;
- (2) $\mathbb{H}_V \subset \mathbb{H}$ — множество хостов в сети, имеющих уязвимости, позволяющие нарушителю получить права пользователя или администратора;
- (3) $H = \|\mathbb{H}\|$ — число хостов в анализируемой сети;
- (4) $H_V = \|\mathbb{H}_V\|$ — число хостов, на которые нарушитель может перейти при реализации атакующих действий;
- (5) V — число уязвимостей во внутренней базе данных уязвимостей;
- (6) A_h — количество уязвимостей на хосте $h \in \mathbb{H}_V$, позволяющих нарушителю получить права пользователя или администратора и перейти на данный хост;
- (7) $A_{\max} = \max_{h \in \mathbb{H}_V} A_h$ — максимальное число уязвимостей по всем хостам анализируемой сети, позволяющих нарушителю получить права пользователя или администратора, перейти на захваченный хост и продолжить реализацию атакующих действий с данного хоста.

Рассмотрим два случая:

- (1) $H_V = H$ (наихудший случай с точки зрения вычислительной сложности алгоритма формирования дерева атак) и
- (2) $H_V \neq H$.

В первом случае нарушитель действует следующим образом: сканирует сеть на предмет выявления функционирующих хостов (обнаруживает все три (H) хоста), реализует последовательно для каждого из обнаруженных хостов все доступные атакующие действия (V) и переходы на захваченные хосты. Тогда сложность будет рассчитываться по следующей рекуррентной формуле:

$$F(H_V) \leq H_V (V + A_{\max} F(H_V - 1)),$$

где первое слагаемое представляет собой число атакующих действий, направленные на все хосты анализируемой сети, второе позволяет определить число атакующих действий с учетом переходов нарушителя на каждый из захваченных хостов.

$$\begin{aligned} F(H_V) &\leq H_V (V + A_{\max} F(H_V - 1)) = \\ &= H_V V + H_V (H_V - 1) A_{\max} (V + A_{\max} F(H_V - 2)) = \\ &= H_V V + H_V (H_V - 1) A_{\max} V + H_V (H_V - 1) A_{\max}^2 F(H_V - 2) = \\ &= H_V V + H_V (H_V - 1) A_{\max} V + H_V (H_V - 1) A_{\max}^2 (H_V - 2) (V + A_{\max} F(H_V - 3)) = \\ &= H_V V + H_V (H_V - 1) A_{\max} V + H_V (H_V - 1) (H_V - 2) A_{\max}^2 V + \\ &+ H_V (H_V - 1) (H_V - 2) A_{\max}^3 F(H_V - 4) = \\ &= \left\{ \prod_{i=1}^k (n-i) = n(n-1)\dots(n-k) = \frac{n(n-1)\dots(n-k)(n-k-1)\dots 1}{(n-k-1)(n-k-2)\dots 1} = \frac{n!}{(n-k-1)!} \right\} = \\ &= \frac{H_V!}{(H_V-1)!} A_{\max}^0 V + \frac{H_V!}{(H_V-2)!} A_{\max}^1 V + \frac{H_V!}{(H_V-3)!} A_{\max}^2 V + \dots = \\ &= V \sum_{i=0}^{H_V-2} A_{\max}^i \frac{H_V!}{(H_V-i-1)!} = V \sum_{i=1}^{H_V-1} A_{\max}^{(i-1)} \frac{H_V!}{(H_V-i)!} \end{aligned}$$

Таким образом, имеем:

$$F(H_V) \leq V \sum_{i=1}^{H_V-1} A_{\max}^{(i-1)} \frac{H_V!}{(H_V-i)!}, \quad H_V = H.$$

Сложность алгоритма формирования дерева атак для *второго случая* будет вычисляться по следующей формуле:

$$\begin{aligned}
 F(H_V) &\leq HV + H_V(V + A_{\max}F(H_V - 1)) = HV + H_VV + H_VA_{\max}F(H_V - 1) = \\
 &= HV + H_VV + H_VA_{\max}(HV + (H_V - 1)(V + A_{\max}F(H_V - 2))) = \\
 &= HV + H_VV + H_VA_{\max}HV + H_V(H_V - 1)A_{\max}(V + A_{\max}F(H_V - 2)) = \\
 &= HV(1 + H_VA_{\max}) + H_VVA_{\max}^0 + H_V(H_V - 1)A_{\max}V + H_V(H_V - 1)A_{\max}^2F(H_V - 2) = \\
 &= HV\left(\frac{H_V!}{(H_V - 0)!}A_{\max}^0 + \frac{H_V!}{(H_V - 1)!}A_{\max}^1\right) + \frac{H_V!}{(H_V - 1)!}VA_{\max}^0 + \frac{H_V!}{(H_V - 2)!}VA_{\max}^1 + \dots = \\
 &= HV\sum_{i=0}^{H_V-1}A_{\max}^i\frac{H_V!}{(H_V - i)!} + V\sum_{i=1}^{H_V-1}A_{\max}^{(i-1)}\frac{H_V!}{(H_V - i)!} = \\
 &= V\left[H\sum_{i=0}^{H_V-1}A_{\max}^i\frac{H_V!}{(H_V - i)!} + \sum_{i=1}^{H_V-1}A_{\max}^{(i-1)}\frac{H_V!}{(H_V - i)!}\right].
 \end{aligned}$$

Таким образом, имеем:

$$F(H_V) \leq V\left[H\sum_{i=0}^{H_V-1}A_{\max}^i\frac{H_V!}{(H_V - i)!} + \sum_{i=1}^{H_V-1}A_{\max}^{(i-1)}\frac{H_V!}{(H_V - i)!}\right], \quad H_V \neq H.$$

Сложность *алгоритма анализа дерева атак* равна сложности алгоритма обхода графа в глубину, оцениваемая как

$$O(V + E),$$

где V — число вершин,

E — число ребер дерева атак.

Для практической оценки сложности формирования дерева атак были проведены эксперименты с несколькими сетями.

Результаты данных экспериментов приведены на рис. 22.

Приведенные выше расчеты показывают, что сложность предложенного подхода к анализу защищенности компьютерных сетей растет пропорционально факториалу H_V .

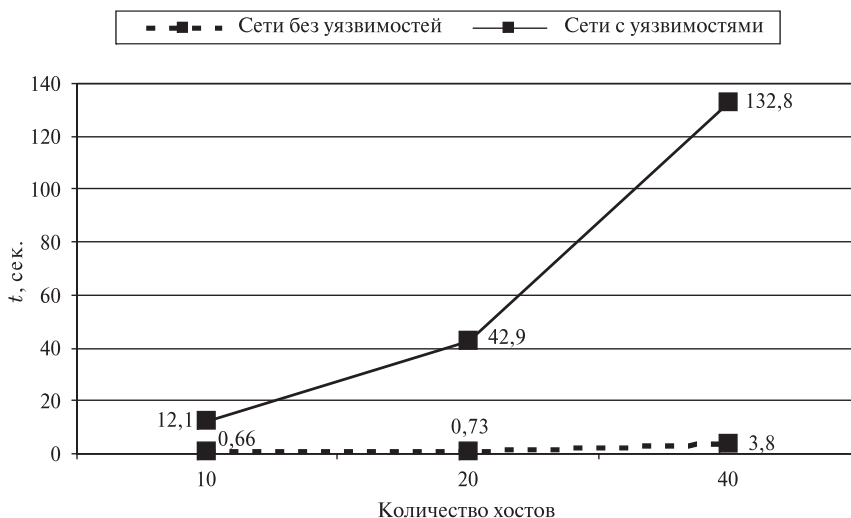


Рис. 22. Общее время анализа защищенности тестовых компьютерных сетей

Представленный подход может быть применен для небольших сетей, но его использование для больших сетей без соответствующей модификации проблематично.

Для уменьшения сложности в работе предлагается использовать следующие подходы:

- (1) разбиение сети на фрагменты, распараллеливание вычислений для каждого фрагмента и последующее объединение решений;
- (2) использование агрегирования и абстрагирования представлений атакующих действий и (или) объектов сети (по атакующим действиям; по объектам сети; комбинирование различных подходов к использованию агрегирования и абстрагирования);
- (3) комбинирование распараллеливания, агрегирования и абстрагирования.

Заключение

Основным результатом работы является разработка оригинального подхода к анализу защищенности компьютерных сетей, предназначенного для использования как на этапах их проектирования, так и эксплуатации. Подход базируется на автоматической генерации общего графа атак, отра-

жающего возможные распределенные сценарии атак (с учетом конфигурации сети, реализуемой политики безопасности, а также местоположения, целей, уровня знаний и стратегий злоумышленника), и использовании комплекса разнообразных метрик безопасности, характеризующих защищенность компьютерной сети с различной степенью детализации и с учетом различных аспектов.

Основными компонентами предлагаемой в работе системы анализа защищенности является модуль формирования внутреннего представления анализируемой сети и политики безопасности, модуль генерации общего графа атак и модуль анализа защищенности, базирующийся на разработанных таксономиях метрик защищенности. Важным компонентом системы является структурированное хранилище информации, объединяющее группы различных баз данных и знаний, в том числе о сети, реализуемой политике безопасности и атакующих действиях.

Система анализа защищенности (САЗ), построенная на основе предложенного подхода, была реализована программно, и с ней проведены эксперименты на тестовой компьютерной сети.

Предлагаемый подход обладает рядом особенностей, которые подчеркивают его новизну и практическую значимость:

- использование для анализа защищенности комплекса различных моделей, построенных на экспертных знаниях, в том числе моделей злоумышленника, многоуровневых моделей сценариев атак, формирования графа атак, расчета метрик защищенности и определения общего уровня защищенности;
- учет разнообразия местоположения, целей, уровня знаний и сценариев злоумышленника;
- использование при построении общего графа атак не только параметров конфигурации компьютерной сети, но и правил реализуемой политики безопасности; возможность оценки влияния на степень защищенности указанных данных;
- учет как собственно атакующих действий (по использованию уязвимостей), так и обычных действий легитимного пользователя и действий по разведке, которые нарушитель может реализовать при получении определенных полномочий на скомпрометированных хостах;
- возможность исследования различных угроз безопасности (нарушения конфиденциальности, целостности, доступности, получения информации о сети, получения прав локального пользователя и администратора) для различных ресурсов сети;

- возможность определения «узких мест» (хостов, ответственных за большее количество трасс атак и уязвимостей, имеющих наиболее высокую возможность компрометации);
- возможность задания запросов к системе вида «что если» («what-if»), например, какова будет защищенность при изменении определенного параметра конфигурации сети, правила политики безопасности, введении знаний о новой уязвимости и т. п.;
- применение для построения графа атак актуализированных баз данных об уязвимостях;
- использование для расчета части первичных метрик защищенности подхода «CVSS. Common Vulnerability Scoring System» [52];
- использование для вычисления метрик защищенности качественных методик анализа риска (в частности модифицированных методики оценки серьезности сетевой атаки, предложенной в SANS/GIAC, и методики FRAP [54]).

Система анализа защищенности, использующая предложенный подход, предназначена для функционирования на различных этапах жизненного цикла компьютерной сети, включая этапы проектирования и эксплуатации. На этапе проектирования САЗ оперирует с моделью анализируемой компьютерной сети, которая базируется на заданной спецификации компьютерной сети и реализуемой политики безопасности. На этапе эксплуатации САЗ взаимодействует с реальной компьютерной сетью.

В результате анализа защищенности определяются уязвимости, строятся трассы (графы) возможных атак, выявляются «узкие места» в компьютерной сети, и вычисляются различные метрики безопасности, которые могут быть использованы для оценки общего уровня защищенности компьютерной сети (системы), а также уровня защищенности ее компонентов.

Полученные результаты обеспечивают выработку обоснованных рекомендаций по устранению выявленных узких мест и усилению защищенности системы. На основе данных рекомендаций пользователь вносит изменения в конфигурацию реальной сети или в ее модель, а затем, если необходимо, повторяет процесс анализа уязвимостей и оценки уровня защищенности. Таким образом, обеспечивается требуемый уровень защищенности компьютерной сети (системы) на всех этапах ее жизненного цикла.

Направлениями дальнейших исследований является совершенствование моделей компьютерных атак и оценки уровня защищенности, в частности системы метрик защищенности и правил их вычисления, развитие компонентов САЗ, модификация подхода для анализа защищенности больших

сетей и проведение дальнейшей экспериментальной оценки предложенных решений.

Работа выполнена при финансовой поддержке РФФИ (проект № 07-01-00547), программы фундаментальных исследований ОИТВС РАН (контракт № 3.2/03), Фонда содействия отечественной науке и при частичной финансовой поддержке, осуществляемой в рамках проектов Евросоюза POSITIF (контракт IST-2002-002314) и RE-TRUST (контракт № 021186-2).

Литература

1. *Alberts C., Dorofee A.* Managing Information Security Risks: The OCTAVE Approach. Addison Wesley Professional, 2002.
2. *Chapman C., Ward S.* Project Risk Management: Processes, Techniques and Insights. Chichester, John Wiley, 2003.
3. *Peltier T. R., Peltier J., Blackley J. A.* Managing a Network Vulnerability Assessment. Auerbach Publications, 2003.
4. *Петренко С. А., Симонов С. В.* Управление информационными рисками. Экономически оправданная безопасность. М.: Компания АйТи; ДМК Пресс, 2004.
5. *Астахов А.* Анализ защищенности корпоративных автоматизированных систем // Jet Info, № 7, 2002.
6. *Симонов С. В.* Технологии и инструментарий для управления рисками // JetInfo, № 2, 2003.
7. *Черешкин Д. С., Кононов А. А., Бурдин О. А.* Комплексная экспертная система «АванГард» как средство управления рисками нарушения информационной безопасности // Научно-техническая информация, Сер. 2. 2000, № 12. С. 15–28.
8. *Кононов А. А., Поликарпов А. К.* Автоматизация построения профилей защиты с использованием комплексной экспертной системы «АванГард» // Научно-техническая информация, Сер. 1, 2003, № 8. С. 27–32.
9. *Skroch M., McHugh J., Williams JM.* Information Assurance Metrics: Prophecy, Process or Pipedream // Panel Workshop. National Information Systems Security Conference (NISSC-2000). 2000.
10. *Evans S., Bush S., Hershey J.* Information Assurance through Kolmogorov Complexity // DARPA Information Survivability Conference and Exposition (DISCEX-H-2001). 2001.
11. *Swanson M.* Security Metrics Guide for Information Technology Systems. National Institute of Standards and Technology Special Publication, No. 800–26, 2001.
12. *Swanson M., Nadya B., Sabato J., Hash J., Graffo L.* Security Metrics Guide for Information Technology Systems. National Institute of Standards and Technology Special Publication, No. 800–55, 2003.
13. *Vaughn R., Henning R., Siraj A.* Information Assurance Measures and Metrics: State of Practice and Proposed Taxonomy // Proceedings of 36th Hawaii International Conference on System Sciences (HICSS-03), 2003.
14. *Schneider B.* Attack Trees // Dr. Dobb's Journal, Vol. 12, 1999.

15. *Gorodetski V., Kutenko I.* Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool // Lecture Notes in Computer Science. Springer Verlag, Vol. 2516, 2002.
16. *Kumar S., Spafford E. H.* An Application of Pattern Matching in Intrusion Detection. Technical Report CSDTR 94 013. Purdue University, 1994.
17. *Chung M., Mukherjee B., Olsson R. A., Puketza N.* Simulating Concurrent Intrusions for Testing Intrusion Detection Systems // Proceeding of the 1995 National Information Systems Security Conference. 1995.
18. *Cohen F.* Simulating Cyber Attacks, Defenses, and Consequences // IEEE Symposium on Security and Privacy, Berkeley, CA, 1999.
19. *Yuill J., Wu F., Settle J., Gong F.* Intrusion-detection for incident-response, using a military battlefield-intelligence process // Computer Networks, No. 34, 2000.
20. *Dawkins J., Campbell C., Hale J.* Modeling network attacks: Extending the attack tree paradigm // Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, Johns Hopkins University, 2002.
21. *Iglun K., Kemmerer R. A., Porras P. A.* State Transition Analysis: A Rule-Based Intrusion Detection System // IEEE Transactions on Software Engineering, Vol. 21, No. 3. 1995.
22. *Chi S.-D., Park J. S., Jung K.-C., Lee J.-S.* Network Security Modeling and Cyber Attack Simulation Methodology // Lecture Notes in Computer Science. Springer Verlag, Vol. 2119, 2001.
23. *Shepard B., Matuszek C., Fraser C. B., etc.* A Knowledge-based approach to network security: applying Cyc in the domain of network risk assessment // The Seventeenth Innovative Applications of Artificial Intelligence Conference (IAAI-05), 2005.
24. *Jha S., Linger R., Longstaff T., Wing J.* Survivability Analysis of Network Specifications // International Conference on Dependable Systems and Networks, IEEE CS Press, 2000.
25. *Jha S., Sheyner O., Wing J.* Minimization and reliability analysis of attack graphs. Technical Report CMU-CS-02-109. Carnegie Mellon University, 2002.
26. *Lye K., Wing J.* Game Strategies in Network Security // International Journal of Information Security, February, 2005.
27. *Ritchey R. W., Ammann P.* Using model checking to analyze network vulnerabilities. IEEE Symposium on Security and Privacy, 2000.
28. *Sheyner O., Haines J., Jha S., Lippmann R., Wing J. M.* Automated generation and analysis of attack graphs. Proceedings of the 2002 IEEE Symposium on Security and Privacy, pages 254–265, 2002.
29. *Sheyner O.* Scenario Graphs and Attack Graphs // CMU Computer Science Department technical report CMU-CS-04-122, Ph. D. dissertation, April 2004.
30. *Wing J. M.* Scenario Graphs Applied to Security // Proceedings of Workshop on Verification of Infinite State Systems with Applications to Security, Timisoara, Romania, March 2005.
31. *Rothmaier G., Krumm H.* A Framework Based Approach for Formal Modeling and Analysis of Multi-level Attacks in Computer Networks // Formal Techniques for Networked and Distributed Systems — FORTE 2005: 25th IFIP WG 6.1 International Conference. Lecture Notes in Computer Science, Vol. 3731, pp. 247–260, 2005.
32. *Singh S., Lyons J., Nicol D. M.* Fast Model-based Penetration Testing // Proceedings of the 2004 Winter Simulation Conference, 2004.

33. Swiler L., Phillips C., Ellis D., Chakerian S. Computer-attack graph generation tool // DIS-CEX '01, 2001.
34. Hariri S., Qu G., Dharmagadda T., Ramkishore M., Raghavendra C. S. Impact Analysis of Faults and Attacks in Large-Scale Networks // IEEE Security & Privacy, September/October, 2003.
35. Rieke R. Tool based formal Modelling, Analysis and Visualisation of Enterprise Network Vulnerabilities utilising Attack Graph Exploration // EICAR 2004. Conference Proceedings. 2004.
36. Dantu R., Loper K., Kolan P. Risk Management using Behavior based Attack Graphs // Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04). 2004.
37. Ou X., Govindavajhala S., Appel A. W. MulVAL: A Logic-based Network Security Analyzer // 14th Usenix Security Symposium, August 2005.
38. Noel S., Jacobs M., Kalapa P., Jajodia S. Multiple coordinated views for network attack graphs // IEEE Workshop on Visualization for Computer Security (VizSEC2005), Minneapolis, MN, October, 2005.
39. Noel S., Jajodia S. Managing attack graph complexity through visual hierarchical aggregation // Proc. ACM Workshop on Visualization and Data Mining for Computer Security, October 2004.
40. Noel S., Jajodia S. Understanding complex network attack graphs through clustered adjacency matrices // Proc. 21st Annual Computer Security Conference (ACSAC), Tucson, AZ, December 5–9, 2005.
41. Swarup V., Jajodia S., Pamula J. Rule-based topological vulnerability analysis // Proc. 3rd Int'l. Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS 2005), Springer Lecture Notes in Computer Science, Vol. 3685, 2005.
42. Kotenko I., Stepashkin M., Ulanov A. Agent-based modeling and simulation of malefactors' attacks against computer networks // Security and Embedded Systems. D. N. Serpanos, R. Giladi (Eds.). IOS Press. 2006. P. 139–146.
43. Котенко И. В., Степашкин М. В., Богданов В. С. Архитектуры и модели компонентов активного анализа защищенности на основе имитации действий злоумышленников // Проблемы информационной безопасности. Компьютерные системы. 2006, № 2, С. 7–24.
44. Котенко И. В., Степашкин М. В., Богданов В. С. Анализ защищенности компьютерных сетей на различных этапах их жизненного цикла // Изв. вузов. Приборостроение. Т. 49, № 5, 2006, С. 3–8.
45. Котенко И. В., Степашкин М. В. Метрики безопасности для оценки уровня защищенности компьютерных сетей на основе построения графов атак // Защита информации. Инсайд, № 3, 2006. С. 36–45.
46. Kotenko I., Stepashkin M. Network Security Evaluation based on Simulation of Malefactor's Behavior // SECRIPT 2006. International Conference on Security and Cryptography. Proceedings. Portugal. 7–10 August 2006. P. 339–344.
47. Kotenko I., Stepashkin M. Analyzing network security using malefactor action graphs // IJCSNS International Journal of Computer Science and Network Security, VOL. 6 No. 6, June 2006. P. 226–235. ISSN: 1738-7906.
48. Kotenko I., Stepashkin M. Attack Graph based Evaluation of Network Security // The 10th IFIP Conference on Communications and Multimedia Security. CMS'2006. Heraklion, Greece. 19–21 October 2006. 2006. Proceedings. Lecture Notes in Computer Science, Vol. 4237, 2006. P. 216–227.

-
49. Котенко И. В., Степашкин М. В., Богданов В. С. Оценка безопасности компьютерных сетей на основе графов атак и качественных метрик защищенности // Труды СПИИ-РАН, Выпуск 3, Том 2. СПб.: Наука, 2006. С. 30–49.
 50. NVD. National Vulnerability Database. <http://nvd.nist.gov>
 51. OSVDB: The Open Source Vulnerability Database. <http://www.osvdb.org>
 52. CVSS. Common Vulnerability Scoring System. <http://www.first.org/cvss>
 53. NVD-Severity. National Vulnerability Database Severity Ranking.
<http://nvd.nist.gov/cvss.cfm>
 54. FRAP. Facilitated Risk Analysis Process. <http://www.peltierassociates.com>
 55. Netfilter/iptables documentation. URL: <http://www.netfilter.org/documentation/>