САНКТ-ПЕТЕРБУРГСКИЙ ФЕДЕРАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
РОССИЙСКОЙ АКАДЕМИИ НАУК

Е.В. Дойникова, И.В. Котенко

ОЦЕНИВАНИЕ ЗАЩИЩЕННОСТИ И ВЫБОР КОНТРМЕР ДЛЯ УПРАВЛЕНИЯ КИБЕРБЕЗОПАСНОСТЬЮ

Москва
2021

УДК 378
ББК 22.1д
Д62

Д62 Оценивание защищенности и выбор контрмер для управления кибербезопасностью: монография / Е. В. Дойникова и И. В. Котенко. — М.: РАН, 2021. — 184 с.

ISBN 978-5-907366-23-7

Вопросы оценивания защищенности и выбора контрмер являются существенными для управления кибербезопасностью и становятся все актуальнее с распространением Интернета вещей. Хотя существуют стандарты и руководящие документы, содержащие требования к оцениванию защищенности и выбору контрмер, а также большое количество изданий, посвященных вопросам криптографии, в настоящий момент не существует монографий и учебных пособий, содержащих детальный анализ методик оценивания защищенности и выбора контрмер для управления кибербезопасностью, способов формирования и вычисления метрик, необходимых для обеспечения требований по оцениванию защищенности, а также представленных на рынке средств, реализующих данные методики. Поэтому существует необходимость в подготовке и выпуске такой монографии для исследователей и специалистов в области аудита и управления кибербезопасностью, а также разработчиков систем управления кибербезопасностью.

В предлагаемой монографии «Оценивание защищенности и выбор контрмер для управления кибербезопасность» рассматривается широкий круг вопросов, охватывающих как научные задачи разработки методик, моделей и алгоритмов оценивания защищенности и выбора контрмер, так и задачи их практического применения в системах управления кибербезопасностью. Кроме того, в монографии подробно рассматриваются существующие стандарты оценивания защищенности и выбора контрмер, а также представления данных безопасности. Содержимое монографии является результатом многолетних научных исследований и практических разработок авторов в области оценивания защищенности и выбора контрмер для управления кибербезопасностью.

Книга предназначена для специалистов, занимающихся разработкой методов и средств защиты информации в компьютерных системах и сетях, студентов и аспирантов, научных сотрудников и преподавателей по дисциплинам, связанным с информационной безопасностью, а также для широкого круга читателей.

Работа выполнена при финансовой поддержке Гранта РНФ № 21–71–20078 в Санкт-Петербургском Федеральном исследовательском центре Российской академии наук.

ISBN 978-5-907366-23-7

Содержание

Список сокращений

ИБ – информационная безопасность
ИС – информационная система
ИТ – информационные технологии
ИТТ – информационные и телекоммуникационные технологии
КС – компьютерная сеть
МЭ – межсетевой экран
НСД – несанкционированный доступ
ОС – операционная система
ПД – персональные данные
ПО – программное обеспечение
СМИБ – система менеджмента информационной безопасности
СОА – сервис-ориентированная архитектура
СОВ – система обнаружения вторжений
СОЗВК – система оценивания защищенности и выбора контрмер
СПВ – системы предотвращения вторжений
ФСТЭК – Федеральная служба по техническому и экспортному контролю
ALE – ожидаемые годовые потери (Annual Loss Expectancy)
ARF – формат отчета об активах (Asset Reporting Format)
CAPEC – общее перечисление и классификация шаблонов атак (Common Attack Pattern Enumeration and Classification)
CCE – общее перечисление конфигураций (Common Configuration Enumeration)
CCSS – общая система оценки конфигураций (Common Configuration Scoring System)
CCTA – центральное компьютерное и телекоммуникационное агентство (Central Computer and Telecommunications Agency)
CPE – общее перечисление платформ (Common Platform Enumeration)
CRE – общее перечисление защитных мер (Common Remediation Enumeration)
CVE – общий словарь уязвимостей и дефектов (Common Vulnerabilities and Exposures)
CVSS – общая система оценки уязвимостей (Common Vulnerability Scoring System)
CWE – общее перечисление слабых мест (Common Weakness Enumeration)
DNS – система доменных имен (Domain Name System)
ERI – расширенная информация по защитным мерам (Extended Remediation Information)
FIPS – Федеральные стандарты обработки информации (Federal Information Processing Standards)
FIRST – Форум групп безопасности и реагирования на инциденты (Forum of Incident Response and Security Teams)

FISMA – Федеральный акт управления информационной безопасностью (Federal Information Security Management Act)
FRAAP – облегченный процесс анализа рисков (Facilitated Risk Analysis and Assessment Process)
GTADM – теоретико-игровая модель защиты от атак (Game Theoretical Attack-Defense Model)
HIPAA – Акт о мобильности и подотчётности медицинского страхования (Healthcare Insurance Portability and Accountability Act)
HRCM – иерархическая модель вычисления рисков (Hierarchical Risk Computing Model)
LAFE – оценка локальной годовой частоты (Local Annual Frequency Estimate)
NCP – процент компрометации сети (Network Compromise Percentage)
NIST – Национальный институт стандартов и технологий (National Institute of Standards & Technology)
NVD – Национальная база уязвимостей (National Vulnerability Database)
NVP – чистая текущая стоимость (Net Present Value)
NSA – Национальное агентство безопасности (National Security Agency)
OCIL – открытый язык отображения проверок безопасности (Open Checklist Interactive Language)
OpenSKE – разработанные открытые знания в области безопасности (Open Security Knowledge Engineered)
OCTAVE – оперативная оценка критических угроз, активов и уязвимостей (Operationally Critical Threat, Asset, and Vulnerability Evaluation)
OVAL – открытый язык спецификации уязвимостей и оценки (Open Vulnerability and Assessment Language)
PDCA – «Планирование-Выполнение-Проверка-Действие» (Plan-Do-Check-Act)
RASQ – относительный коэффициент поверхности атаки (Relative Attack Surface Quotient)
ROI – возврат инвестиций (Return on Investment)
RORI – возврат инвестиций в реагирование (Return-On-Response-Investment)
SAFE – оценка стандартной годовой частоты (Standard Annual Frequency Estimate)
SCAP – протокол автоматизации управления данными безопасности (Security Content Automation Protocol)
SEM – управление событиями безопасности (Security Event Management)
SIEM – управление информацией и событиями безопасности (Security Information and Event Management)
SIM – мониторинг IP адреса источника (Source IP Address Monitoring)

SWID – идентификация программного обеспечения (Software Identification)
TMSAD – модель доверия для данных автоматизации безопасности (Trust Model for Security Automation Data)
WASC – консорциум по безопасности веб-приложений (Web Application Security Consortium)
XML – расширяемый язык разметки (eXtensible Markup Language)
XCCDF – расширяемый формат описания списков контроля конфигураций (eXtensible Configuration Checklist Description Format)

Список таблиц

Список рисунков

Введение

В век информационных технологий особенно актуальными становятся вопросы управления кибербезопасностью. Одними из основных задач управления кибербезопасностью являются оценивание защищенности и выбор контрмер. За последние годы появилось большое количество стандартов, исследований и программных продуктов в этой области. Цель данной монографии состоит в том, чтобы дать читателю общее представление как о процессах оценивания защищенности и выбора контрмер при управлении кибербезопасностью в целом, так и о существующих подходах, методах и методиках их осуществления.

Ключевым понятием при оценивании защищенности и выборе контрмер является понятие риска. С момента первого упоминания термина «риск», его понимание претерпело значительные изменения. В [1, 2] описана эволюция понятия «риск». Первые попытки научного исследования риска на основе теории вероятностей были связаны с развитием азартных игр, результаты позднее стали использоваться для управления страховыми рисками (16–17 вв.). Новый толчок развитию теории риска дало обоснование соблюдения в будущем прошлых закономерностей в связи с ограниченностью доступных для анализа данных (18–19 вв., Г.В. Лейбниц, Я. Бернулли). Затем был введен учет последствий риска (1738 г., Д. Бернулли), и появилось представление о возможном ущербе как плате за риск в связи с развитием теории предпринимательского риска (18 в., А. Смит, Д.С. Милль, Д.М. Кларк, К. Маркс). Позднее наметилось различие между условиями, вероятность которых может быть рассчитана, и условиями, вероятность которых непредсказуема, то есть условиями неопределенности (1850 г., Й. фон Тюнен), что развилось в понимание риска как измеримой неопределенности (1921 г., Ф. Найт). Новый виток в развитии теории предпринимательского риска связан с расширением понятия риска посредством учета издержек риска (1883–1946 гг., Дж.М. Кейнс). В 1900–1960 гг. риск рассматривается как результат воздействия антропогенных и природных факторов, и появляется необходимость системного подхода к управлению рисками, а также системы оценивания и прогнозирования риска.

Риски кибербезопасности за рубежом стали рассматриваться в середине 20 в. в связи с развитием применения компьютерных технологий [3], и, как следствие, ростом киберпреступности. В Россию термин информационный риск пришел в конце 20 в. [4]. Согласно ISO/IEC 27005 [5] под риском понимается влияние неопределенности на цели. В области кибербезопасности целью является обеспечение конфиденциальности, целостности, доступности, а также неотказуемости, подотчетности, аутентичности и достоверности. Их нарушение может привести к серьезным неблагоприятным последствиям для организаций разного рода, поскольку они все больше полагаются на информационные технологии [6]. Поэтому возникает необходимость управления кибербезопасностью или рисками информационной безопасности. Как отмечено в [7], «либо вы управляете рисками, либо риски управляют вами».

В ответ на обозначенную потребность появилось большое количество стандартов, систем и методик в области кибербезопасности. К основным

зарубежным стандартам в области кибербезопасности относятся стандарты FIPS (Federal Information Processing Standards) и NIST (National Institute of Standards & Technology), а также стандарты ISO (группы ISO 27*), касающиеся систем управления информационной безопасностью. В России к основным стандартам относятся руководящие документы ФСТЭК и ГОСТы серии 13335, 15408, 18045, 19791, 20004, 24760, 27*, 29100, 29115, 29128, 29151, 54581, 54582, 54583, 54628. Также необходимо отметить тенденцию к накоплению и стандартизации представления данных безопасности, в том числе уязвимостей, слабых мест программно-аппаратного обеспечения, шаблонов атак, программно-аппаратного обеспечения, конфигураций, контрмер и других. В связи с чем появились проколы управления данными безопасности (например, SCAP [8]), стандарты представления данных (в том числе, CVE для отображения уязвимостей [9], CAPEC для отображения шаблонов атак [10, 11], CPE для отображения программно-аппаратного обеспечения [12] и другие), и базы данных безопасности (например, база уязвимостей NVD [13], база тактик, техник и методов противодействия кибератакам MITRE ATT&CK [14]). Это, в свою очередь, дало толчок к трансформации от ручных методик и средств оценивания защищенности и выбора контрмер к автоматизированным. Методики оценивания защищенности и выбора контрмер отличаются применяемыми моделями и методами. Методики, рассматриваемые в данной монографии, в той или иной степени учитывают возможность использования уязвимостей программного и аппаратного обеспечения в компьютерных атаках и влияние их использования в атаках на активы организаций. Целью оценивания защищенности является выявление тех уязвимостей и слабых мест, которые могут привести к серьезным последствиям для целей организации. Это позволит как выбрать необходимые контрмеры, так и обосновать затраты на обеспечение кибербезопасности. Доступное представление затрат на обеспечение кибербезопасности с обоснованием эффективности и возможных потерь могут дать показатели защищенности. Поэтому подход к управлению кибербезопасностью должен основываться на надежных, выражаемых количественно показателях. Показатели защищенности являются количественными индикаторами атрибутов безопасности информационной системы или технологии, предоставляя способ измерения качества продуктов или сервисов и улучшения эффективности реализуемых процессов. Нельзя улучшить защищенность, не измерив ее. В текущих исследованиях предлагается большое количество различных показателей и методик их вычисления. Это представляется авторам еще одним важным аспектом в области оценивания защищенности и выбора контрмер, который рассматривается в данной монографии.

Большую проблему с точки зрения обеспечения кибербезопасности представляют распределенные сети организаций с большим количеством компьютеров, ввиду огромного количества информации и событий безопасности, которые необходимо обработать. По данным [15], среднее количество сообщений, генерируемых различными устройствами за день, может достигать от нескольких десятков (приложения, прокси серверы, системы контроля доступа, системы обнаружения вторжений и реагирования на вторжения) и сотен (межсетевые экраны, маршрутизаторы, центральные процессоры)

тысяч до нескольких миллионов и более (базы данных). Обработать такое количество информации вручную практически невозможно. Кроме того, в случае если система подвергается атаке, важным становится временной аспект. Следовательно, возникает проблема автоматизированной обработки информации, ее представления в удобном для оператора виде и автоматизированного выбора контрмер. В рамках решения данной проблемы управления кибербезопасностью на текущий момент активно развиваются системы мониторинга безопасности и управления инцидентами (Security Information and Event Management, SIEM). Данные системы предназначены для мониторинга информационной безопасности организации за счет сбора, нормализации, корреляции и агрегации информации безопасности и представления ее в удобном для анализа виде. Реализуемые в них методы оценивания защищенности также представляют интерес на текущий момент.

К сожалению, несмотря на прилагаемые усилия, рынок киберпреступности в России и в мире продолжает расти. По отчету международной компании по предотвращению и расследованию киберпреступлений Group-IB за 2020–2021 год [16], потери только от программ-вымогателей за конец 2019 и 2020 гг. составили более одного миллиарда долларов, рынок продажи доступов в скомпрометированные сети компаний составил более шести миллионов долларов, объем рынка продаж краденых данных банковских карт составил около двух миллиардов долларов, существенно выросло количество фишинг-ресурсов. Поэтому существует необходимость исследования и создания новых методик оценивания защищенности и выбора контрмер, которые позволят улучшить ситуацию в области кибербезопасности в России и мире. Авторами был разработан собственный подход к оцениванию защищенности и выбору контрмер для защиты от кибератак, который приводится в конце данной монографии.

В заключение отметим, что данная работа посвящена вопросам оценивания защищенности и выбора контрмер в распределенных компьютерных системах разного рода, порождаемых угрозами атак на основе использования уязвимостей программно-аппаратного обеспечения, с применением автоматизированных методик и средств.

В первой главе определяется место и роль мониторинга защищенности компьютерных систем, оценивания защищенности и выбора контрмер в процессе управления кибербезопасностью. Для этого подробно рассмотрены основные международные стандарты и стандарты РФ в области кибербезопасности, и представлена таксономия стандартов. Также описаны системы управления кибербезопасностью и место компонентов оценивания защищенности и выбора контрмер в их составе. Во второй главе рассматриваются основные источники данных, применяемые для оценивания защищенности и выбора контрмер, а также основные протоколы управления данными безопасности и стандарты их представления. В третьей главе описываются модели, методики, показатели и средства, используемые для оценивания защищенности. В четвертой главе рассматриваются модели, методики, показатели и средства, применяемые для выбора контрмер. И, наконец, в пятой главе описываются разработанные авторами методики и средства оценивания защищенности и выбора контрмер, основанные на графах атак и зависимостей сервисов.

Глава 1. Место и роль оценивания защищенности и выбора контрмер в процессе управления кибербезопасностью

1.1. Основные стандарты в области информационной безопасности

Для определения места и роли оценивания защищенности и выбора контрмер в процессе управления кибербезопасностью ниже рассмотрены основные международные и российские стандарты в области информационной безопасности (ИБ).

С того момента, как впервые была затронута тема безопасности информации, появилось большое количество стандартов, как российских, так и международных. В области международных стандартов можно выделить стандарты ISO и стандарты NIST. В области российских стандартов следует выделить документы ФСТЭК, методические рекомендации ФСБ России, ГОСТы РФ, банковские стандарты и другие. При этом международные стандарты ISO и ГОСТы РФ коррелируют между собой. Полный список национальных стандартов в области обеспечения информационной безопасности можно найти на сайте ФСТЭК России [17]. Наиболее важные из этих стандартов для темы, рассматриваемой в данной книге, и их взаимосвязи показаны на рисунке 1.1. На рисунке в круглых скобках указана последняя действующая на текущий момент версия международного стандарта, российский аналог международного стандарта выделен цветом, стрелки указывают на наличие ссылки на другой стандарт в рамках рассматриваемого. Более подробную информацию по взаимосвязям стандартов группы 27* можно найти в стандарте ГОСТ Р ИСО/МЭК 27000–2012 [18].

Ниже подробнее рассмотрены основные действующие стандарты РФ в области управления ИБ, затрагивающие вопросы оценивания защищенности и выбора контрмер в общем процессе разработки и функционирования компьютерных систем и сетей, для определения места и роли этих процессов.

Стандарты ГОСТ Р ИСО/МЭК 15408-1-2012 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель» [19], ГОСТ Р ИСО/МЭК 15408-2-2013 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2. Функциональные требования безопасности» (соответствует ISO/IEC 15408-2) [20] и ГОСТ Р ИСО/МЭК 15408–3 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности» [21] определяют требования к функциональным возможностям продуктов информационных технологий (ИТ) и применяемым к ним мерам доверия. Стандарты предоставляют потребителям профиль защиты, разработчикам — задание по безопасности (включающее описание активов, угроз и контрмер), а оценщи-

кам — критерии безопасности, позволяющие определить, соответствует ли объект оценки предъявляемым требованиям безопасности.

Стандарты ГОСТ Р 54581–2011 «Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 1. Обзор и основы» [22], ГОСТ Р 54582–2011 «Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 2. Методы доверия» [23] и ГОСТ Р 54583–2011 «Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 3. Анализ методов доверия» [24] предназначены для выбора методов и средств обеспечения доверия для безопасности информационных и коммуникационных технологий.

Стандарт ГОСТ Р 57628–2017 «Информационная технология. Методы и средства обеспечения безопасности. Руководство по разработке профилей защиты и заданий по безопасности» [25] предназначен для разработки профилей защиты и заданий по безопасности в соответствии со стандартами ГОСТ Р ИСО/МЭК 15408.

ГОСТ Р ИСО/МЭК 18045–2013 «Информационная технология. Методы и средства обеспечения безопасности. Методология оценки безопасности информационных технологий» [26] посвящен оценке безопасности информационных технологий.

ГОСТ Р ИСО/МЭК 19791–2008 «Информационная технология. Методы и средства обеспечения безопасности. Оценка безопасности автоматизированных систем» [27] расширяет стандарты ИСО/МЭК 15408 для учета особенностей автоматизированных систем.

ГОСТ Р ИСО/МЭК 58143–2018 «Информационная технология. Методы и средства обеспечения безопасности. Детализация анализа уязвимостей программного обеспечения в соответствии с ГОСТ Р ИСО/МЭК 15408 и ГОСТ Р ИСО/МЭК 18045. Часть 2. Тестирование проникновения» [28] детализирует стандарты ИСО/МЭК 15408 и 18045 с точки зрения идентификации, выбора и оценки потенциальных уязвимостей программного обеспечения.

Стандарт ГОСТ Р ИСО/МЭК 27031–2012 [29] рассматривает принципы готовности информационно-коммуникационных технологий к обеспечению непрерывности бизнеса при возникновении событий, инцидентов и нарушений (в том числе безопасности), которые могут влиять на критические функции бизнеса.

Стандарты ГОСТ Р ИСО/МЭК 27033-1-2011 «Информационная технология. Методы и средства обеспечения безопасности. Безопасность сетей. Часть 1. Обзор и концепции» [30] и ГОСТ Р ИСО/МЭК 27033-3-2014 «Информационная технология. Методы и средства обеспечения безопасности. Безопасность сетей. Часть 3. Эталонные сетевые сценарии. Угрозы, методы проектирования и вопросы управления» [31] посвящены вопросам анализа и обеспечения безопасности сетей.

Вопросы безопасности приложений отражены в стандарте ГОСТ Р ИСО/МЭК 27034-1-2014 «Информационная технология. Методы и средства обеспечения безопасности. Безопасность приложений. Часть 1. Обзор и общие понятия» [32].

Вопросы обеспечения конфиденциальности персональных данных рассматриваются в стандарте ГОСТ Р ИСО/МЭК 29100–2013 «Информационная технология. Методы и средства обеспечения безопасности. Основы обеспечения приватности» [33].

В стандарте ГОСТ Р ИСО/МЭК 27001–2006 «Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования» [34] представлены требования и основные этапы разработки систем менеджмента информационной безопасности организаций, независимо от типа, масштаба и сферы их деятельности.

В стандарте ГОСТ Р ИСО/МЭК 27002–2012 «Информационная технология. Методы и средства обеспечения безопасности. Свод норм и правил менеджмента информационной безопасности» [35] представлены основные нормы и правила менеджмента информационной безопасности.

В стандарте ГОСТ Р ИСО/МЭК 13335-1-2006 «Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий» [36] приведены общие понятия и модели управления безопасностью информационных и телекоммуникационных технологий (ИТТ) в организации, описаны основные компоненты безопасности, вовлеченные в процесс управления кибербезопасностью, и их связи.

В стандарте ГОСТ Р ИСО/МЭК 27003–2012 «Информационная технология. Методы и средства обеспечения безопасности. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Руководство по реализации системы менеджмента информационной безопасности» [37] рассматриваются вопросы определения, разработки и внедрения систем менеджмента информационной безопасности в соответствии с ГОСТ Р ИСО/МЭК 27001–2006.

ГОСТ Р ИСО/МЭК 27004–2011 «Информационная технология. Методы и средства обеспечения безопасности. Менеджмент информационной безопасности. Измерения» [38] содержит рекомендации по оценке эффективности системы менеджмента информационной безопасности и мер и средств контроля. Полученные измерения должны помочь в усовершенствовании систем менеджмента информационной безопасности и, как следствие, улучшении безопасности организации.

ГОСТ Р ИСО/МЭК 27005–2010 «Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности» [39] представляет собой руководство по менеджменту рисков информационной безопасности.

Стандарты ГОСТ Р ИСО/МЭК 27006–2008 «Информационная технология. Методы и средства обеспечения безопасности. Требования к органам, осуществляющим аудит и сертификацию систем менеджмента информационной безопасности»[1] [40], ГОСТ Р ИСО/МЭК 27007–2014 «Информационная технология. Методы и средства обеспечения безопасности. Руководства по аудиту систем менеджмента информационной безопасности» [41] и ГОСТ Р 56045–2014 «Информационная технология. Методы и средства

[1] Планируется введение новой версии ГОСТ Р ИСО/МЭК 27006–2020 с 2021-07-01.

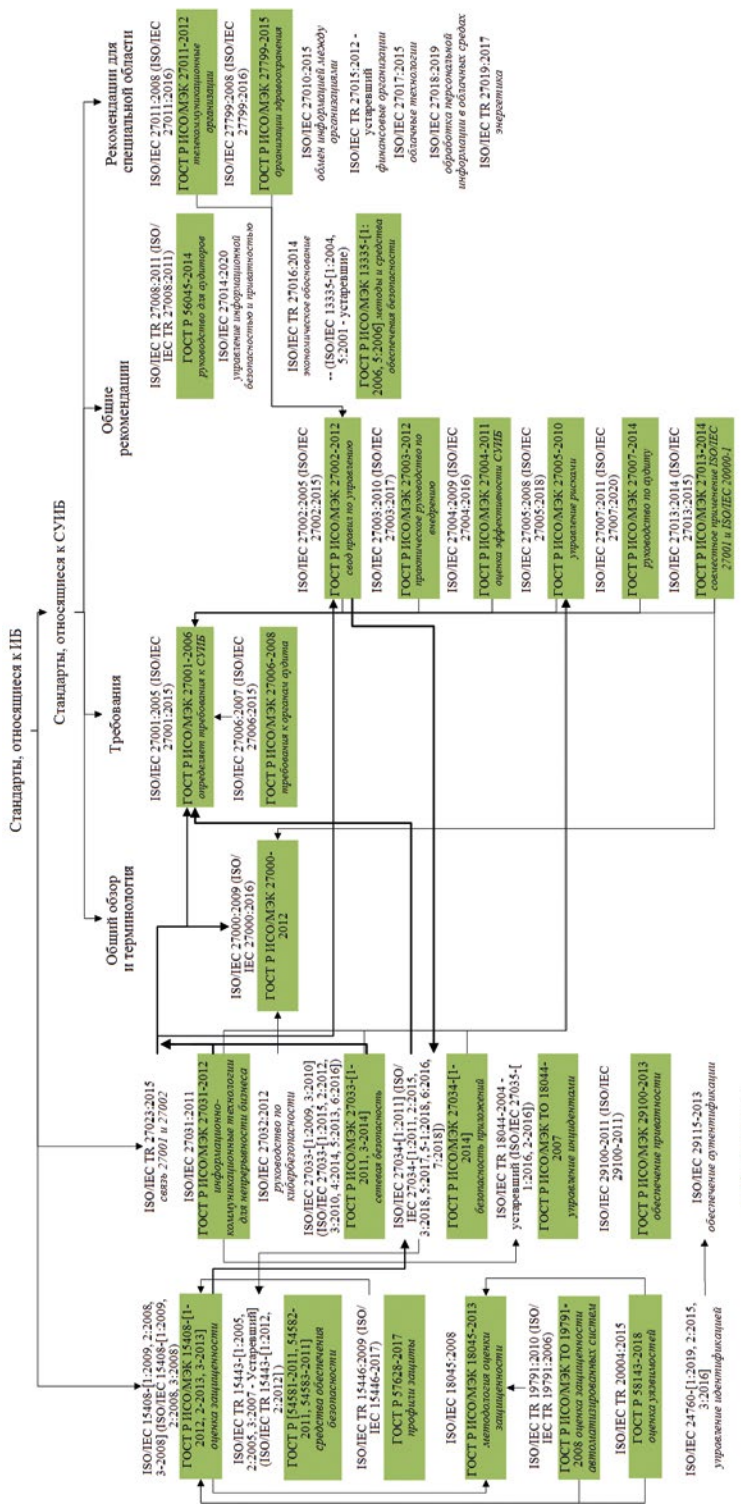


Рис. 1.1. Соответствие между международными стандартами кибербезопасности ISO и ГОСТами РФ

обеспечения безопасности. Рекомендации для аудиторов в отношении мер и средств контроля и управления информационной безопасностью» [42] затрагивают вопросы аудита систем менеджмента информационной безопасности организаций, в том числе устанавливают критерии для органов, осуществляющих аудит и сертификацию, содержат руководства по менеджменту программы аудита систем менеджмента информационной безопасности (СМИБ) и проверке мер и средств контроля и управления информационной безопасностью.

Стандарт ГОСТ Р ИСО/МЭК 27013–2014 «Информационная технология. Методы и средства обеспечения безопасности. Руководство по совместному использованию стандартов ИСО/МЭК 27001 и ИСО/МЭК 20000-1» [43] представляет собой руководство по совместному использованию стандартов ИСО/МЭК 27001 и ИСО/МЭК 20000–1, то есть учитывает информационные активы при менеджменте информационной безопасности.

Стандарты ГОСТ Р ИСО/МЭК 27011–2012 «Руководства по менеджменту информационной безопасности для телекоммуникационных организаций на основе ИСО/МЭК 27002» [44] и ГОСТ Р ИСО 27799–2015 «Информатизация здоровья. Менеджмент защиты информации в здравоохранении по ИСО/МЭК 27002» [45] предоставляют руководства по менеджменту информационной безопасности для организаций разного типа (телекоммуникационных и здравоохранения).

ФСТЭК России был разработан ряд методических (Таблица 1.1) и руководящих (Таблица 1.2) документов в области информационной безопасности, которые, в свою очередь, соответствуют национальным стандартам РФ.

Таблица 1.1

Методические документы ФСТЭК России

Название	Затрагиваемые вопросы	Дата утверждения	Соответствующие требования	Соответствующие ГОСТы
Профили защиты операционных систем типов «Б» и «В» [46]	Определяет взаимосвязи требований и функций безопасности операционных систем (ОС) [47]	11 мая 2017 г.	Требования безопасности информации к ОС, утвержденные приказом ФСТЭК России от 19 августа 2016 г. № 119	ГОСТ Р ИСО/МЭК 15408
Профили защиты ОС типа «А» [48]	Определяет взаимосвязи требований и функций безопасности ОС [49]	8 февраля 2017 г.	Требования безопасности информации к ОС, утвержденные приказом ФСТЭК России от 19 августа 2016 г. № 119	ГОСТ Р ИСО/МЭК 15408
Профили защиты межсетевых экранов (МЭ) [50]	Определяет взаимосвязи требований и функций безопасности МЭ [51]	12 сентября 2016 г.	Требования к МЭ, утвержденные приказом ФСТЭК России от 9 февраля 2016 г. № 9	ГОСТ Р ИСО/МЭК 15408

Название	Затрагиваемые вопросы	Дата утверждения	Соответствующие требования	Соответствующие ГОСТы
Профили защиты средств контроля съемных машинных носителей информации [52]	Определяет взаимосвязи требований к функциям безопасности средств контроля съемных машинных носителей информации [53]	1 декабря 2014 г.	Требования к средствам контроля съемных машинных носителей информации, утвержденные приказом ФСТЭК России от 28 июля 2014 г. № 87	ГОСТ Р ИСО/МЭК 15408
Меры защиты информации в государственных информационных системах [54]	Рассматривает вопросы выбора организационных и технических защитных мер, направленных на обеспечение конфиденциальности информации, целостности информации и доступности информации, для государственных информационных систем разных классов требований защиты информации, а также содержание мер защиты информации [54]	11 февраля 2014 г.	Требования о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах, утвержденные приказом ФСТЭК России от 11 февраля 2013 г. № 17	—
Профили защиты средств доверенной загрузки [55]	Определяет взаимосвязи требований к функциям безопасности средств доверенной загрузки [56]	30 декабря 2013 г.	Требования к средствам доверенной загрузки, утвержденные приказом ФСТЭК России от 27 сентября 2013 г. № 119	ГОСТ Р ИСО/МЭК 15408
Профили защиты средств антивирусной защиты [57]	Определяет взаимосвязи требований к функциям безопасности средств антивирусной защиты [58]	14 июня 2012 г.	Требования к средствам антивирусной защиты, утвержденные приказом ФСТЭК России от 20 марта 2012 г. № 28	ГОСТ Р ИСО/МЭК 15408
Профили защиты систем обнаружения вторжений (СОВ) [59]	Определяет взаимосвязи требований к функциям безопасности СОВ пятого и шестого классов защиты [60]	6 марта 2012 г.	Требования к системам обнаружения вторжений, утвержденные приказом ФСТЭК России от 6 декабря 2011 г. № 638	ГОСТ Р ИСО/МЭК 15408
Профили защиты СОВ [61]	Определяет взаимосвязи требований к функциям безопасности СОВ четвертого класса защиты [62]	3 февраля 2012 г.	Требования к системам обнаружения вторжений, утвержденные приказом ФСТЭК России от 6 декабря 2011 г. № 638	ГОСТ Р ИСО/МЭК 15408

Название	Затрагиваемые вопросы	Дата утверждения	Соответствующие требования	Соответствующие ГОСТы
Базовая модель угроз безопасности персональных данных (ПД) при их обработке в информационных системах ПД [63]	Описывает угрозы безопасности ПД при их обработке в информационных системах ПД [63]	15 февраля 2008 г.	—	—
Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах ПД [64] (устарела)	Описывает методику определения угроз безопасности ПД для обеспечения их безопасности при обработке [64]	14 февраля 2008 г.	—	—
Методика оценки угроз безопасности информации	Описывает методику определения антропогенных угроз безопасности информации в системах и сетях [65]	5 февраля 2021 г.	Подпункт 4 пункта 8 Положения о ФСТЭК, утвержденного Указом Президента РФ от 16 августа 2004 г. № 1085	—

Таблица 1.2

Руководящие документы ФСТЭК России

Название	Затрагиваемые вопросы	Дата утверждения	Соответствующие ГОСТы или РД
Безопасность информационных технологий. Критерии оценки безопасности ИТ [66]	Описывает требования к безопасности ИТ, в том числе определяет виды требований, содержит каталог требований безопасности и каталог требований доверия к безопасности [66]	19 июня 2002 г.	ГОСТ Р ИСО/МЭК 15408-2002
Защита от несанкционированного доступа (НСД) к информации Часть 1. Программное обеспечение средств защиты информации Классификация по уровню контроля отсутствия недекларированных возможностей [67]	Классификация средств защиты информации по уровню контроля отсутствия недекларированных возможностей [67]	4 июня 1999 г.	ГОСТ 19.202–78 ГОСТ 19.402–78 ГОСТ 19.502–78 ГОСТ 19.404–79 ГОСТ 19.401–78
Средства вычислительной техники. МЭ. Защита от НСД к информации. Показатели защищенности от НСД к информации [68]	Классификация МЭ по уровню защищенности от НСД [68]	25 июля 1997 г.	—

Название	Затрагиваемые вопросы	Дата утверждения	Соответствующие ГОСТы или РД
Автоматизированные системы. Защита от НСД к информации. Классификация автоматизированных систем и требования по защите информации [69]	Классификация автоматизированных систем, подлежащих защите от НСД, и требования безопасности к соответствующим автоматизированным системам [69]	30 марта 1992 г.	Дополняет ГОСТ 34.003–90, ГОСТ 34.601-90
Защита от НСД к информации. Термины и определения [70]	Термины и определения в области защиты от НСД [70]	30 марта 1992 г.	—
Концепция защиты средств вычислительной техники и автоматизированных систем от НСД к информации [71]	Принципы защиты информации от НСД [71]	30 марта 1992 г.	—
Руководство по разработке ПЗ и заданий по безопасности [72]	Руководство по разработке ПЗ и заданий по безопасности [72]	2003 г.	РД Гостехкомиссии России «Критерии оценки безопасности информационных технологий»
Безопасность информационных технологий. Руководство по формированию семейств ПЗ [73]	Определяет порядок формирования семейств ПЗ для изделий информационных технологий [73]	2003 г.	РД Гостехкомиссии России «Критерии оценки безопасности информационных технологий» РД «Безопасность информационных технологий. Руководство по разработке ПЗ и заданий по безопасности» РД «Безопасность информационных технологий. Руководство по регистрации ПЗ» РД «Безопасность информационных технологий. Положение по разработке ПЗ и заданий по безопасности»
Безопасность информационных технологий. Руководство по регистрации ПЗ [74]	Определяет процедуры формирования, ведения и использования реестра ПЗ [74]	2003 г.	РД Гостехкомиссии России «Критерии оценки безопасности информационных технологий» ISO/IEC 15292–2001 (устарел)
Безопасность информационных технологий. Положение по разработке ПЗ и заданий по безопасности [75]	Определяет «порядок разработки, оценки, регистрации и публикации ПЗ и заданий по безопасности для продуктов и систем ИТ, предназначенных для обработки информации, отнесенной к информации ограниченного доступа» [75]	—	ГОСТ Р ИСО/МЭК 15408–2002 (устарел, заменен на 15408-1-2012, 15408-2-2013, 15408-3-2013) РД «Критерии оценки безопасности ИТ» РД «Безопасность ИТ. Руководство по разработке семейств ПЗ» РД «Безопасность ИТ. Руководство по разработке ПЗ и заданий по безопасности» РД «Безопасность ИТ. Руководство по регистрации ПЗ»

Название	Затрагиваемые вопросы	Дата утверждения	Соответствующие ГОСТы или РД
Защита информации. Специальные защитные знаки. Классификация и общие требования [76]	Классификация защитных знаков для контроля доступа к объектам защиты [76]	25 июля 1997 г.	—
Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от НСД в автоматизированных системах и средствах вычислительной техники [77]	Устанавливает порядок исследований и разработок в области защиты информации и создания средств защиты информации от НСД [77]	30 марта 1992	—

1.2. Определение места и роли оценивания защищенности и выбора контрмер

Для определения места и роли оценивания защищенности и выбора контрмер в процессе управления кибербезопасностью ниже подробнее рассмотрены некоторые из перечисленных государственных стандартов.

Согласно ГОСТ Р ИСО/МЭК 27001–2006 [34] СМИБ основана на использовании методов оценивания бизнес-рисков для разработки, внедрения, функционирования, мониторинга, анализа, поддержки и улучшения ИБ. Основные этапы разработки СМИБ согласно ГОСТ Р ИСО/МЭК 27001–2006 представлены на рисунке 1.2. Рамкой выделены этапы, связанные с оцениванием защищенности на основе анализа и оценки рисков, а также обработкой рисков путем выбора контрмер.

Базисом для СМИБ служит система управления рисками, так как управление рисками позволяет ответить на четыре вопроса управления безопасностью: что защищать, почему, от чего (на эти вопросы отвечает оценка защищенности на основе анализа и оценки рисков) и как защищаться (обработка рисков путем выбора контрмер) [7]. Очевидно, что для решения задач оценивания защищенности на основе анализа и оценки рисков, а также обработки рисков путем выбора контрмер, необходимо учитывать этапы определения подхода к оценке риска в организации и идентификации рисков, так как они предоставляют необходимые входные данные и, следовательно, определяют требования к разрабатываемому подходу. На рисунке 1.3 более подробно рассмотрен состав данных этапов (соответствующих этапам оценивания защищенности и выбора контрмер).

В соответствии с описываемым стандартом, определение подхода к оценке риска, включая определение методологии оценки риска, идентификация рисков, анализ и оценка рисков, а также обработка рисков, являются необходимыми этапами внедрения управления безопасностью в организации. Кроме того, отмечено, что на этапе анализа функционирующей СМИБ, необходимо способствовать обнаружению событий ИБ и предотвращать инциденты ИБ, то есть осуществлять мониторинг ИБ. Однако данный стандарт не дает непосредственного описания методологий и носит организационный характер.

Рис. 1.2. Основные этапы разработки СМИБ согласно [34]

Рис. 1.3. Состав этапов управления информационной безопасностью, связанных с оцениванием защищенности и выбором контрмер

Описание методологии оценки рисков более подробно приведено в серии стандартов ГОСТ Р ИСО/МЭК 13335.

На рисунке 1.4 представлены основные компоненты безопасности, вовлеченные в процесс управления безопасностью ИТТ, и их связи согласно стандарту ГОСТ Р ИСО/МЭК 13335-1-2006 [36]. К ним относятся: активы, угрозы, уязвимости, воздействия, контрмеры и риск.

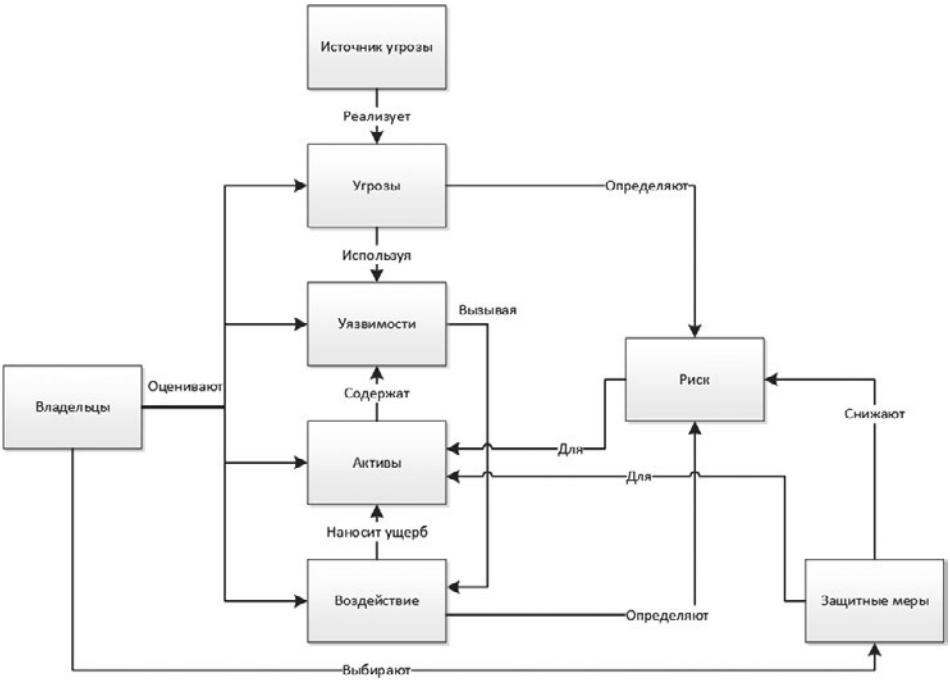


Рис. 1.4. Основные компоненты безопасности и их связи согласно ГОСТ Р ИСО/МЭК 13335-1-2006 [36]

Активы (материальные активы, информация, программное обеспечение, способность производить продукт или предоставлять услугу, люди, нематериальные ресурсы) несут ценность для организации. Эта ценность определяется владельцами активов. Кроме того, активы могут иметь *уязвимости*, которые могут снизить ценность активов и влияют на требования к защите активов. Для повышения безопасности активов применяются *защитные меры* (или контрмеры).

Угрозы (неавторизованное разрушение, раскрытие, модификация, порча, недоступность или потеря) наносят ущерб активам посредством использования уязвимостей. В зависимости от источника, выделяются угрозы среды и угрозы, обусловленные человеческим фактором (случайные или целенаправленные). К характеристикам угрозы относятся: источник (внутренний или внешний), мотивация, частота возникновения, правдоподобие, вредоносное воздействие.

По определению, *воздействие* — это результат инцидента ИБ, вызванного угрозой и нанесшего ущерб ее активу (разрушение конкретного актива, повреждение ИТТ, нарушение их конфиденциальности, целостности, доступности, неотказуемости, подотчетности, аутентичности и достоверности).

Контроль над воздействием позволяет достичь равновесия между предполагаемыми последствиями инцидента и стоимостью защитных мер, при этом следует учитывать вероятность возникновения инцидента.

Риск определяется как способность конкретной угрозы использовать уязвимость одного или нескольких активов для нанесения ущерба организации. Таким образом, риск характеризуется комбинацией двух факторов: вероятностью возникновения инцидента и его разрушительным воздействием. На риск может повлиять изменение активов, угроз, уязвимостей или защитных мер. Таким образом, методология оценивания защищенности на основе оценки риска должна включать учет таких изменений. Обработка риска включает в себя устранение, снижение, перенос и принятие риска. *Защитные меры* включают в себя действия, процедуры и механизмы, способные обеспечить безопасность от возникновения угрозы, уменьшить уязвимость, ограничить воздействие инцидента в системе безопасности, обнаружить инциденты и облегчить восстановление активов.

Данный стандарт дает представление о том, какие компоненты необходимо учесть при разработке методологии оценки рисков. Однако в нем не предлагается конкретных методологий оценки рисков.

Стандарт [38] рассматривает вопросы измерения эффективности СМИБ, а также их место в общем цикле «Планирование-Выполнение-Проверка-Действие» (PDCA). На этапе планирования осуществляется оценивание вероятности нарушения безопасности, с учетом угроз, уязвимостей и их последствий, связанных с активами, а также выбор защитных мер. На этапе выполнения внедряются защитные меры, выбираются способы измерения их результативности и проводится измерение результативности. На этапе проверки проводится анализ результативности СМИБ, на основе которого на этапе действия реализуются улучшения СМИБ. К целям измерений, связанных с ИБ, в контексте СМИБ относятся: оценивание эффективности реализованных мер и средств контроля и управления или их групп; оценивание эффективности реализованной СМИБ; верификация степени, до которой были удовлетворены установленные требования безопасности; содействие повышению результативности ИБ с точки зрения общих рисков основной деятельности организации; предоставление сведений для проверки, проводимой руководством с целью содействия принятию решений, касающихся СМИБ, и обоснования необходимых улучшений в реализованной СМИБ. С точки зрения оценивания защищенности и выбора контрмер к целям измерений относятся следующие из этих целей: верификация степени, до которой были удовлетворены установленные требования безопасности; содействие повышению результативности ИБ с точки зрения общих рисков основной деятельности организации. То есть учитываются оценка рисков и выбор защитных мер на этапе планирования. В дальнейшем, на этапе осуществления, можно измерять эффективность защитных мер и пересматривать их состав.

ГОСТ Р ИСО/МЭК 27005–2010 [39] описывает основные виды деятельности, связанные с менеджментом риска ИБ, включая установление контекста, оценку риска, обработку риска, принятие риска, коммуникацию риска, и мониторинг и переоценку риска (рисунок 1.5). На стадии установления контекста происходит определение основных критериев, необходимых для менеджмента риска ИБ (критериев оценки риска, критериев влияния (ущерба) и критериев принятия риска).

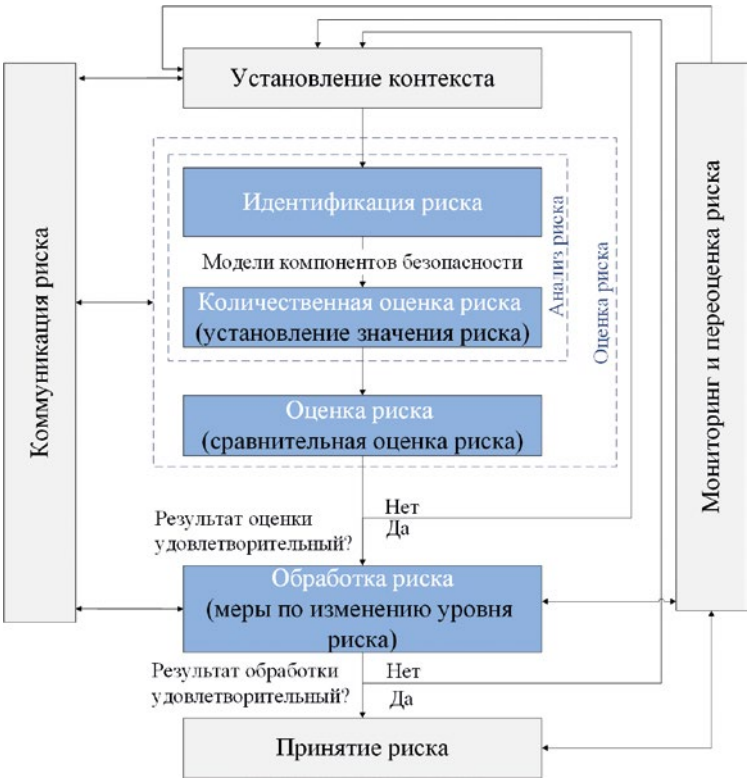


Рис. 1.5. Процесс менеджмента риска информационной безопасности и место оценки и обработки риска

Остановимся подробнее на процессе оценки рисков ИБ. Согласно [39] в процессе оценки риски должны быть идентифицированы, охарактеризованы количественно или качественно, и для них должны быть назначены приоритеты. Как видно из рисунка 1.5, процесс оценки риска включает анализ риска (идентификация риска и установление значения риска) и оценку риска. Он позволяет установить ценность информационных активов, выявить потенциальные угрозы и уязвимости, определить существующие меры и средства контроля и управления и их влияние на риски, определить возможные последствия и назначить приоритеты рискам.

Идентификация риска включает определение активов, определение угроз, определение существующих мер и средств контроля и управления, выявление уязвимостей, определение последствий.

Методология установления значения риска может быть качественной (на основе качественной шкалы, например, низкий, средний, высокий), количественной (на основе шкалы с числовыми значениями) или комбинированной.

На этапе обработки риска должны быть выбраны меры и средства контроля и управления риском (для снижения риска, сохранения риска, предотвращения риска или переноса риска). Меры и средства контроля и управления риском выбираются на основе стоимости их реализации, их ожидаемой эффективности и результатов оценки риска.

В стандарте приведены примеры подходов к оценке рисков: высокоуровневая оценка риска, детальная оценка риска, комбинированная оценка риска.

Высокоуровневая оценка риска (базовый подход) включает высокоуровневую оценку последствий и подразумевает применение стандартных защитных мер ко всем системам информационных технологий. Преимуществами такого подхода является минимизация времени и средств на выбор защитных мер. Недостатком данного подхода является применение одинакового базового уровня безопасности для различных систем информационных технологий организации, независимо от их критичности. Такой вариант стратегии подходит для организаций, в которых все используемые информационные технологии характеризуются низким уровнем требований к обеспечению безопасности.

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

Комбинированный подход подразумевает проведение предварительной оценки риска для выделения систем с высоким уровнем риска. К таким системам в дальнейшем применяется детальная оценка риска, а для остальных систем ограничиваются базовым подходом. Преимуществом данного подхода является вложение ресурсов и средств именно в те области, где это наиболее необходимо. Недостатком является возможность ошибочного отнесения системы к тем, которые не требуют детального анализа риска.

Ниже подробнее рассмотрены операции, входящие в детальную оценку рисков.

Для оценки идентифицированных активов стандарт предлагает два возможных способа. Первый подразумевает определение их ценности для организации (например, первоначальная стоимость актива, стоимость его обновления или воссоздания). При этом важно определить однозначные критерии оценки и шкалу оценки (количественную — например, в денежных единицах, или качественную — например, «Высокая»/«Средняя»/«Низкая»). Другой подход основывается на затратах, понесенных по причине утраты конфиденциальности, целостности или доступности (насколько

пострадает деловая деятельность организации и другие активы системы ИТ от утечки, искажения, недоступности и/или разрушения информации). При этом необходимо определить уровни возможного ущерба, связанного с воздействием нежелательного инцидента (например, от нарушения законов, потери репутации, финансовых потерь и т. п.).

Кроме того, следует выделить зависимости одних активов от других. Выходными данными такой операции является список активов и их оценок с учетом конфиденциальности, целостности и доступности информации, а также стоимостью их замены.

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

Оценка уязвимостей включает идентификацию уязвимостей, которые могут быть использованы источниками угроз для нанесения ущерба активам, и оценку вероятного уровня слабости.

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

Оценка рисков включает идентификацию и оценку рисков для выбора обоснованных защитных мер. Величина риска определяется ценностью подвергающихся риску активов, вероятностью реализации угроз, возможностью использования уязвимостей идентифицированными угрозами, а также наличием защитных мер. Метод оценки рисков должен быть повторяемым и прослеживаемым. В стандарте рассматривается несколько табличных методов оценки риска: матрица с заранее определенными значениями; ранжирование угроз по мерам риска; оценка частоты появления и возможного ущерба, связанного с рисками; разграничение между допустимыми и недопустимыми рисками.

Метод «Матрица с заранее определенными значениями» состоит в определении ценности активов (с учетом их важности для личной безопасности, репутации организации, хода работ и т.п.) на основе качественной шкалы со значениями 1–4. Для оценки угроз и уязвимостей используются пары вопросников по каждому типу угрозы для каждой группы активов, позволяющих определить вероятности возникновения угроз и легкость их реализации в уязвимых местах. За каждый ответ начисляются очки, которые накапливаются и сравниваются с рангами, в результате определяются уровни угроз (от высокого до низкого) и уровни уязвимости. Уровни риска определяются по шкале от 1 до 8.

Преимущество метода состоит в возможности ранжирования соответствующих рисков. Недостатки метода — субъективность, ручное определение угроз и уязвимостей.

Метод «Ранжирование угроз по мерам риска» состоит в определении меры риска как произведения ценности актива как ущерба от воздействия

(например, по шкале от 1 до 5) и определении вероятности возникновения угрозы (например, по шкале от 1 до 5) и дальнейшем ранжировании угроз на основе полученных значений риска.

Преимущество метода — ранжирование угроз по мерам рисков. Недостатки метода — субъективность, ручное определение угроз и уязвимостей, задаются приоритеты без действительного значения угрозы.

Метод «Оценка частоты появления и возможного ущерба, связанного с рисками» состоит в определении приоритетных активов на основе оценки воздействия нежелательных событий. Активы оцениваются по уровню ущерба от каждой угрозы. Затем на основе вероятности возникновения угрозы и легкости ее возникновения в уязвимых местах определяют значение частоты. Оценка по активам/угрозам определяется суммированием для каждого актива. Для определения оценки системы складываются оценки всех активов.

Преимущества метода — ранжирование активов по величине ущерба, выделение наиболее приоритетных систем. Недостатки метода — субъективность, ручное определение угроз и уязвимостей, задаются приоритеты без реальных оценок.

Метод «Разграничение между допустимыми и недопустимыми рисками» состоит в ранжировании мер рисков по срочности принятия мер.

Преимущества метода — ранжирование рисков на допустимые и недопустимые риски. Недостатки метода — субъективность, ручное определение угроз и уязвимостей, задаются приоритеты без реальных оценок.

Таким образом, можно заключить, что недостатком всех перечисленных методов является:

— ручной подход, который может привести к упущению важных деталей;

— субъективность оценок;

— качественная шкала оценок, без реальных количественных выражений потерь и рисков.

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

1.3. SIEM-системы как средство управления кибербезопасностью

SIEM-системы предназначены для обработки и анализа событий безопасности из различных источников с целью повышения эффективности управления безопасностью организации. SIEM-системы объединяют функции двух категорий продуктов предыдущего поколения: систем управления информацией по безопасности (Security Information Management, SIM) и систем управления событиями безопасности (Security Event Management, SEM). К функциям SIM относятся: сбор записей о событиях, архивация,

формирование отчетов об истории событий, форензика. К функциям SEM относятся: предоставление отчетов в реальном времени, сбор записей о событиях, нормализация, корреляция, агрегация. Таким образом, базовые функции SIEM-систем включают: сбор записей о событиях из различных источников; нормализацию с целью представления записей о событиях из различных источников в едином формате для связи и анализа событий; корреляцию с целью связи записей о событиях и событий от различных систем и приложений для ускорения обнаружения и реагирования на угрозы безопасности; агрегацию с целью снижения объема данных по событиям путем удаления идентичных записей; составление отчетов с целью представления результатов заинтересованным лицам в реальном времени или в форме долгосрочных отчетов; решение задач форензики.

Путем выполнения данных функций SIEM-системы позволяют повысить эффективность управления безопасностью в организации за счет упрощения процесса анализа событий, их своевременного обнаружения, оперативной обработки и связи между собой разрозненных событий. Кроме того, SIEM-системы позволяют упростить процесс аудита безопасности за счет решения проблемы согласованности информации по безопасности и процесс обнаружения нарушителей за счет учета записей о событиях.

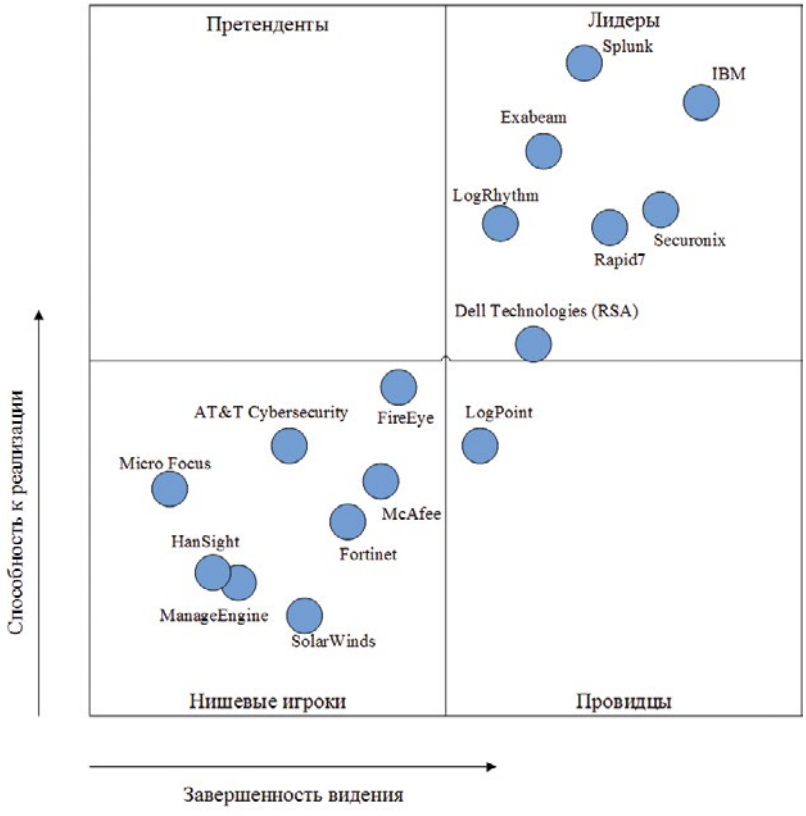


Рис. 1.6. Магический квадрант Gartner в сегменте SIEM-систем, февраль 2020 года [86]

В настоящий момент существует большое количество коммерческих решений в области SIEM-систем. На рисунке 1.6 представлены основные производители SIEM-систем по версии компании Gartner [78] на февраль 2020 года. Лидерами названы: IBM (продукт QRadar SIEM) [79]; Splunk (продукт Splunk Enterprise Security) [80]; Exabeam SIEM [81]; Securonix SIEM [82]; LogRhythm (продукт LogRhythm SIEM) [83]; Rapid7 (продукт InsightIDR) [84]; Dell Technologies (RSA) (продукт NetWitness Platform) [85].

В данных системах могут использоваться различные показатели, отражающие текущую ситуацию по защищенности. Эти показатели обычно характеризуют инциденты безопасности и возможные защитные меры на основе сценариев «ЕСЛИ-ТО», а также различные объекты системы с точки зрения безопасности, в том числе количество уязвимостей, инцидентов и т.п. Такие показатели не дают полной картины рисков кибербезопасности и бизнес-рисков, порождаемых потенциальными угрозами, и не позволяют принять всесторонне обоснованное решение по выбору и внедрению контрмер.

Шаг в сторону расширения возможностей SIEM-систем с точки зрения оценивания защищенности и выбора контрмер был сделан, например, в рамках проекта MASSIF [87, 88]. Для повышения эффективности управления защищенностью системы в архитектуру SIEM-системы был добавлен компонент оценивания защищенности [89, 90].

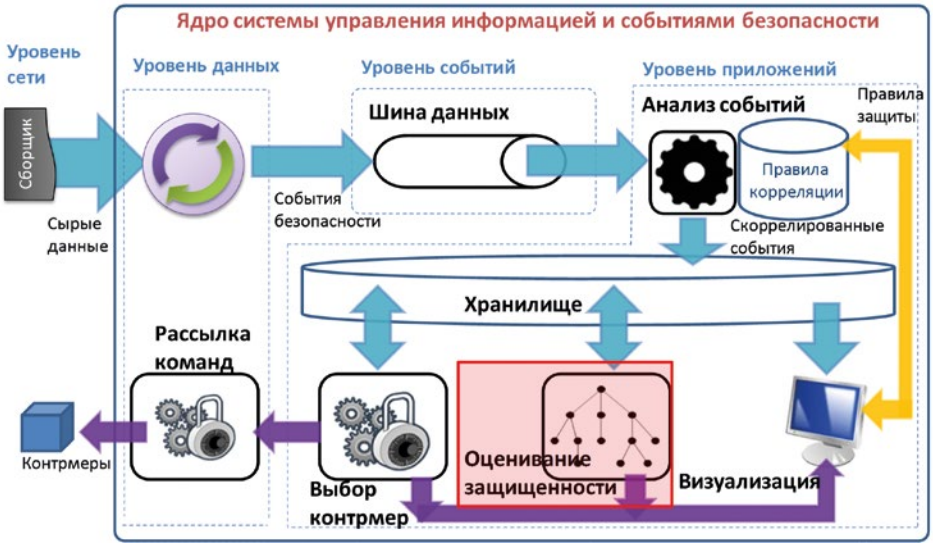


Рис. 1.7. Архитектура SIEM-системы [87]

Архитектура SIEM-системы, разработанной в рамках проекта MASSIF, представлена на рисунке 1.7. Вначале необработанные данные по безопасности в разных форматах с внешних сенсоров попадают в компонент сбора данных. Затем данные о событиях поступают на надежную шину данных, откуда передаются в систему корреляции и обработки, а затем в общее хранилище данных. Далее, скореллированные данные обрабатываются в ком-

понente моделирования атак и оценки рисков и поступают в компонент поддержки принятия решений. Для представления в реальном времени отчетов о текущем состоянии защищенности, правилах корреляции событий и предложениях от системы поддержки принятия решений служит компонент визуализации. Далее решения, предложенные компонентом поддержки принятия решений и утвержденные при необходимости пользователем через компонент визуализации, отправляются в компонент, ответственный за применение этих решений.

Выводы по главе 1

В области менеджмента ИБ существует большое количество стандартов. На основе анализа существующих стандартов можно выделить следующие основные этапы оценивания защищенности на основе оценки риска: установление контекста, оценка риска, обработка риска, принятие риска, коммуникация риска, мониторинг и переоценка риска. Можно сделать вывод, что для эффективной обработки риска предпочтительной является детальная количественная оценка риска, которая включает тщательное определение и установление ценности активов, оценку угроз этим активам и оценку уязвимостей.

Под риском будем понимать меру ущерба от нежелательного события (в идеале выраженного в денежных единицах) и возможности того, что это событие произойдет (в идеале определенную в виде частоты) [91]. Активы — что-то, представляющее ценность. Уязвимости — то, что можно использовать, чтобы скомпрометировать активы. Угроза — что-то или кто-то, кто может использовать уязвимость, чтобы скомпрометировать активы.

Процессы оценивания защищенности на основе оценки риска представляют собой достаточно сложные и трудоемкие процессы, требующие серьезных вложений. Поэтому важной задачей является определение методик количественного оценивания и обработки риска, и автоматизация данных процессов, в том числе за счет унификации спецификации входных данных.

Глава 2. Основные источники данных для оценивания защищенности и выбора контрмер

2.1. Протоколы управления кибербезопасностью

Как было отмечено в главе 1, основными компонентами оценивания защищенности компьютерных систем и сетей являются: активы, источники угроз, угрозы, уязвимости, воздействия, защитные меры и риск. Автоматизация идентификации и оценки данных компонентов является важной задачей, так как это ресурсозатратный и трудоемкий процесс. Для решения данной задачи разрабатываются стандарты унифицированного представления и управления данными по безопасности. Примером является протокол автоматизации управления данными безопасности SCAP (Security Content Automation Protocol) [8, 92].

Протокол SCAP был разработан институтом NIST для решения задач автоматизации оценки защищенности. SCAP является спецификацией, которая объединяет ряд стандартов для унифицированного представления и управления данными по безопасности. Он позволяет составить список используемых в системе платформ и приложений (т.е. идентифицировать активы), задать особенности их конфигурации, неблагоприятно влияющие на защищенность, специфицировать список уязвимостей (т.е. идентифицировать уязвимости системы), оценить неблагоприятное влияние конфигураций и уязвимостей (т.е. определить воздействие), и выявить наиболее критичные уязвимости (т.е. оценить уровень риска).

В общем процессе управления безопасностью SCAP отвечает за анализ управления конфигурациями и анализ уязвимостей. Место, занимаемое SCAP среди других протоколов, представлено на рисунке 2.1.

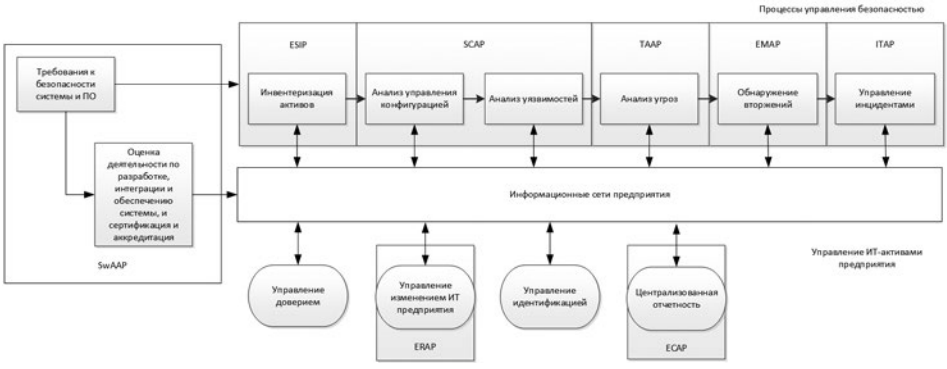


Рис. 2.1. Взаимосвязь протоколов управления безопасностью и место, занимаемое SCAP

Протокол SCAP версии 1.3 [93] включает следующие стандарты:

1. Перечисления, списки или словари, которые задают соглашения по перечислению и именованию, и поддерживаются корпорацией MITRE [94]:

• «Общие уязвимости и дефекты» (Common Vulnerabilities and Exposures, CVE) — список дефектов программного обеспечения (ПО);

• «Общее перечисление платформ» (Common Platform Enumeration, CPE) (версия 2.3) — список и словарь аппаратного обеспечения, операционных систем и приложений;

• «Общее перечисление конфигураций» (Common Configuration Enumeration, CCE) (версия 5) — словарь названий особенностей конфигураций ПО;

• Теги идентификации программного обеспечения (Software Identification, SWID) (версия 2015 г.) — формат отображения идентификаторов ПО и связанных метаданных.

2. Языки спецификации и проверки (определяют способы предоставления инструкций и выражения результатов):

• «Открытый язык спецификации уязвимостей и оценки» (Open Vulnerability and Assessment Language, OVAL) (версия 5.11) — язык отображения информации конфигурации системы, оценки состояния и отчета о результатах оценки, используется для определения соответствия политике безопасности, поддерживается корпорацией MITRE;

• «Расширяемый формат описания списков контроля конфигураций» (eXtensible Configuration Checklist Description Format, XCCDF) (версия 1.2) — XML-спецификация структурированного набора правил конфигурации, используемых операционной системой (ОС), и платформ, служит для определения безопасных конфигураций, поддерживается Агентством национальной безопасности США (National Security Agency, NSA) и NIST;

• «Открытый язык отображения проверок безопасности» (Open Checklist Interactive Language, OCIL) (версия 2.0) — используется для определения соответствия работы системы политике безопасности.

3. Стандарты определения показателей:

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

• «Общая система оценки конфигураций» (Common Configuration Scoring System, CCSS) (версия 1.0) для оценки конфигураций с учетом политик безопасности организации и зависимостей между уязвимостями.

4. Форматы отчетов:

• «Формат отчета об активах» (Asset Reporting Format, ARF) (версия 1.1) — формат представления информации об активах и взаимосвязях между активами и отчетами;

• «Идентификация активов» (Asset Identification) (версия 1.1) — формат уникальной идентификации активов на основе известных идентификаторов и/или известной информации об активах.

5. Целостность:

• «Модель доверия для данных автоматизации безопасности» (Trust Model for Security Automation Data, TMSAD) (версия 1.0) — спецификация

использования цифровых подписей в общей модели доверия, применяемой к другим спецификациям автоматизации безопасности.

На рисунке 2.2 представлена связь между вопросами, возникающими при управлении данными безопасности, и стандартами, входящими в состав SCAP.

Рис. 2.2. Перечень стандартов, входящих в состав SCAP

В таблице 2.1 приведены основные особенности и преимущества протокола SCAP. Они позволяют судить о применимости данного протокола для автоматизации и стандартизации представления данных для расчета показателей в рамках данного исследования.

В последующих разделах подробнее рассмотрены стандарты спецификации данных по безопасности, входящие в SCAP, а также стандарты, находящиеся вне его области действия.

Таблица 2.1

Особенности и преимущества протокола SCAP

Особенность	Преимущество
Стандартизует то, как компьютеры обмениваются информацией об уязвимостях	Обеспечивает взаимодействие продуктов и сервисов различных производителей
Стандартизует то, какой информацией об уязвимостях обмениваются компьютеры	Обеспечивает возможность использования для продуктов и сервисов различных производителей Снижает зависимость от контекста
Базируется на открытых стандартах	Позволяет использовать совместные решения для принятия и развития различных решений Применим для большого числа вариантов использования
Использует стандарты управления конфигурациями и активами	Позволяет использовать информацию об активах и конфигурациях для управления уязвимостями

Особенность	Преимущество
Применим для множества различных методологий управления рисками	Снижает время, усилия и затраты на процесс управления рисками
Содержит детальные ссылки на множество руководящих документов по безопасности	Позволяет автоматизировать процедуры демонстрации соответствия требованиям и подготовки отчетов Снижает возможности недопонимания между руководством/аудиторами и операторами
Основан на стандарте NIST SP 800-53	Позволяет автоматизировать процедуры демонстрации соответствия руководящим документам, например, Federal Information Security Management Act (FISMA), и формирования отчетности [95]

2.2. Системы оценивания уязвимостей

Можно выделить три базовых группы методов оценивания уязвимостей [96]:

— качественное ранжирование — данная группа методов основана на использовании нескольких «качественных» категорий уязвимостей (например, категорий «низкий», «средний» или «высокий»);

— количественное ранжирование — данная группа методов основана на использовании количественной шкалы оценки уязвимостей;

— применение комплексных показателей оценивания уязвимостей.

Примеры реализованных методов и систем оценивания каждой категории представлены на рисунке 2.3.

Наиболее популярной в настоящее время системой оценивания уязвимостей является CVSS.

Стандарт CVSS. Стандарт CVSS оценивает уязвимости по шкале со значениями от 0 до 10, используя комбинацию трех комплексных показателей оценки, которые, в свою очередь, генерируются на основе заданного набора базовых показателей [97].

Рис. 2.3. Методы и системы оценивания уязвимостей

Показатели базовой оценки *BaseScore* (или базовые показатели) задают обязательные характеристики уязвимости. Показатели временной оценки *TemporalScore* (или временные показатели) определяют элементы уязвимости, которые изменяются со временем. Показатели контекстной оценки

EnvironmentalScore (или контекстные показатели) описывают результат использования уязвимости в сети определенной организации.

Для получения оценки по CVSS вначале подсчитываются показатели базовой оценки, позволяющие определить фундаментальные характеристики уязвимости. При их подстановке в базовое уравнение получается оценка в пределах от 0 до 10. На основе базовых показателей также создается вектор, содержащий значения каждого показателя. Далее базовая оценка может быть уточнена установлением значений показателей временной и контекстной оценки (рисунок 2.4). Это позволяет предоставить дополнительный контекст для уязвимости, с более точным отражением риска, который представляет уязвимость [98].

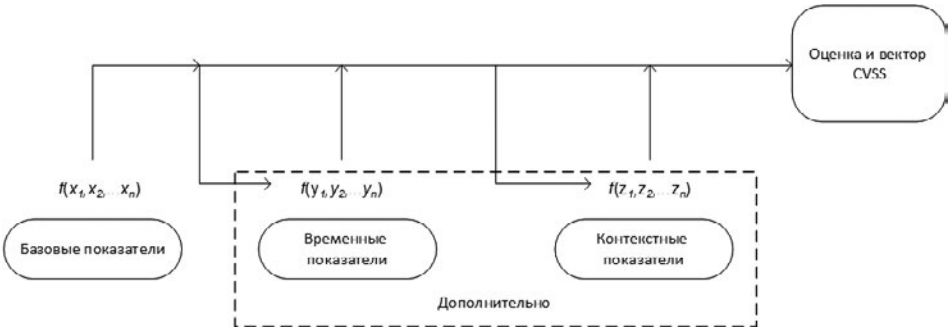


Рис. 2.4. Последовательность оценки CVSS [97]

В группу базовых показателей входят: *вектор доступа* (*Access Vector*, *AV*), *сложность доступа* (*Access Complexity*, *AC*) и *аутентификация* (*Authentication*, *Au*) — определяют способ доступа к уязвимости и нужны ли дополнительные условия для ее эксплуатации; *влияние на конфиденциальность* (*Confidentiality Impact*, *C*), *влияние на целостность* (*Integrity Impact*, *I*), *влияние на доступность* (*Availability Impact*, *A*) — степень потери конфиденциальности, целостности и доступности, измеряют как уязвимость, которая напрямую повлияет на актив ИТ в случае эксплуатации.

Вектор доступа определяет, как эксплуатируется уязвимость (чем более удаленный нарушитель может атаковать хост, тем выше оценка уязвимости). Если уязвимость может быть использована несколькими способами, то выбирается наиболее удаленный доступ.

Сложность доступа определяет сложность атаки, которую необходимо провести для эксплуатации уязвимости после того, как нарушитель получил доступ к системе. Чем ниже сложность, тем выше оценка уязвимости.

Аутентификация определяет, сколько раз атакующий должен аутентифицироваться в системе, чтобы использовать уязвимость (сложность процесса не учитывается, только количество). Чем меньше количество аутентификаций, тем выше оценка. Данный показатель отличается от вектора доступа, то есть считается, что доступ к системе уже есть (кроме логина нужно предоставить еще дополнительную аутентификацию).

Влияние на конфиденциальность определяет ущерб конфиденциальности в результате успешной эксплуатации уязвимости. Увеличение ущерба конфиденциальности увеличивает оценку уязвимости.

Влияние на целостность определяет ущерб целостности после успешной эксплуатации уязвимости. Увеличение ущерба целостности ведет к увеличению оценки уязвимости.

Влияние на доступность определяет ущерб доступности в результате успешной эксплуатации уязвимости. Увеличение ущерба доступности увеличивает оценку уязвимости.

Возможные значения базовых показателей приведены в таблице 2.2.

В группу временных показателей входят: *возможность использования* (*Exploitability*, *E*), *уровень исправления* (*Remediation level*, *RL*), *степень достоверности отчета* (*Report confidence*, *RC*). Данные показатели позволяют учитывать следующие три фактора: доступность кода или методик эксплоита, статус исправления уязвимости и подтверждение технических деталей уязвимости. Если пользователь не хочет учитывать какой-то из временных показателей, есть специальное значение, которое позволяет не учитывать показатель при формировании общей оценки уязвимости.

Возможность использования определяет текущее состояние методик эксплоита или доступности кода. Наличие простого в использовании эксплоита увеличивает количество потенциальных нарушителей. Чем проще использовать уязвимость, тем выше оценка уязвимости.

Таблица 2.2

Значения, принимаемые базовыми показателями CVSS (версия 2.0)

Качественное значение показателя	Описание	Примеры	Численное значение показателя
Вектор доступа (Access Vector)			
Локальный (Local, L)	Физический доступ к уязвимой системе или локальная учетная запись (при этом привилегии учетной записи не учитываются)	Firewire/USB DMA атаки, локальное расширение привилегий (sudo)	0,395
Локальная сеть (Adjacent network, A)	Доступ к широковещательному или коллизионному домену (т.е. доступ к среде передачи данных внутри локальной сети)	Локальная IP подсеть, Bluetooth, IEEE 802.1 и локальный сегмент Ethernet	0,646
Сетевой (Network, N)	Сетевой доступ к уязвимой системе (атакующему не нужен локальный доступ или доступ к локальной сети)	RPC buffer overflow	1,0
Сложность доступа (Access Complexity)			
Высокая (High, H)	Есть специальные условия доступа	— требуются повышенные привилегии или требуется обмануть дополнительные системы; — атака зависит от методов социальной инженерии; — уязвимая конфигурация очень редко встречается на практике; — состояние гонки (race condition)	0,35

Качественное значение показателя	Описание	Примеры	Численное значение показателя
Средняя (Medium, M)	Условия доступа частично специализированы	— атакуемая сторона ограничена группой систем или пользователей; — необходимо собрать дополнительную информацию; — уязвимая конфигурация не является конфигурацией по умолчанию; — необходимо применить социальную инженерию	0,61
Низкая (Low, L)	Нет специальных условий	— продукту требуется доступ к широкому классу систем и пользователей; — уязвимая конфигурация является конфигурацией по умолчанию; — атака может быть проведена вручную и требует слабых навыков или незначительного сбора дополнительной информации; — состояние гонки легко обходится	0,71
Аутентификация (Authentication)			
Множественная (Multiple, M)	Для эксплуатации уязвимости требуется аутентифицироваться два или более раз, даже если это одна и та же учетная запись.	Аутентификация в ОС дополнительно к учетной записи для доступа к приложению, расположенному на хосте	0,45
Одиночная (Single, S)	Для эксплуатации уязвимости необходимо аутентифицироваться один раз		0,56
Нет (None, N)	Для эксплуатации уязвимости аутентификация не нужна		0,704
Влияние на конфиденциальность (Confidentiality Impact)			
Нет (None, N)	Нет ущерба конфиденциальности системы		0,0
Частичный (Partial, P)	Существенное раскрытие информации (возможен доступ к некоторым системным файлам, однако атакующий не имеет контроля над тем, что получил, или ограничена область потерь)	Уязвимости, которые раскрывают только часть таблиц в базе данных (БД)	0,275
Полный (Complete, C)	Полное раскрытие информации, приводящее к раскрытию всех системных файлов	Нарушитель может читать все системные данные (память, файлы и т.п.)	0,660
Влияние на целостность (Integrity Impact)			
Нет (None, N)	Нет ущерба целостности системы		0,0
Частичный (Partial, P)	Возможно изменение некоторых системных файлов или информации, однако атакующий не имеет контроля над тем, что м.б. изменено, или ограничена область того, на что атакующий может повлиять	Файлы системы или приложений м.б. переписаны или изменены, но либо атакующий не контролирует, какие файлы будут атакованы, либо атакующий может изменять файлы только внутри ограниченной области	0,275
Полный (Complete, C)	Полная компрометация целостности системы	Нарушитель может менять любые файлы целевой системы	0,660

Качественное значение показателя	Описание	Примеры	Численное значение показателя
Влияние на доступность (Availability Impact)			
Нет (None, N)	Нет ущерба доступности системы		0,0
Частичный (Partial, P)	Снижена производительность, или есть перерывы в доступности ресурса	Network-based flood attack, которая позволяет ограниченное количество успешных соединений к Интернет-сервису	0,275
Полный (Complete, C)	Полная остановка атакованного ресурса	Нарушитель может сделать ресурс полностью недоступным	0,660

Уровень исправления определяет наличие методов устранения уязвимости. Обычно, когда уязвимость только опубликована, для нее нет исправлений. Обходные пути или горячие заплатки могут дать временное решение, пока не выпущено официальное решение или обновление. Наличие любого из этих вариантов исправления снижает временную оценку. Чем менее официален вариант исправления, тем выше оценка.

Степень достоверности отчета определяет степень уверенности в том, что уязвимость существует, и убедительность известных технических деталей. Чем более удостоверена уязвимость производителем или другими уважаемыми ресурсами, тем выше оценка уязвимости.

Возможные значения временных показателей приведены в таблице 2.3.

Таблица 2.3

Значения, принимаемые временными показателями CVSS (версия 2.0)

Качественное значение показателя	Описание	Численное значение показателя
Возможность использования (Exploitability)		
Непроверенная (Unproven, U)	Нет кода эксплоита или эксплоит полностью теоретический	0,85
Идея (Proof-of-concept, POC)	Есть общее представление кода эксплоита или демонстрация атаки, не являющаяся практической для большинства систем. Для применения кода или методики в конкретной ситуации требуется существенная модификация опытным нарушителем	0,9
Функциональная (Functional, F)	Есть функциональный код эксплоита. Код работает в большинстве ситуаций, когда существует уязвимость	0,95
Высокая (High, H)	Уязвимость может быть использована функциональным мобильным кодом, или не требуется эксплоита и детали широко известны. Код работает в любой ситуации или активно доставляется мобильным агентом (таким, как вирус или червь)	1,00
Не определено (Not defined, ND)	Назначение этого значения показателю позволит не изменять оценку уязвимости	1,00
Уровень исправления (Remediation level)		
Официальное исправление (Official fix, OF)	Доступно законченное решение производителя	0,87

Качественное значение показателя	Описание	Численное значение показателя
Временное исправление (Temporary fix, TF)	Доступно официальное, но временное решение	0,9
Обходной путь (Workaround, W)	Неофициальное решение не от производителя	0,95
Недоступно (Unavailable, U)	Решения нет, или его невозможно применить	1,00
Не определено (Not defined, ND)	Назначение этого значения показателю позволит не изменять оценку уязвимости	1,00
Степень достоверности отчета (Report confidence)		
Не подтвержден (Unconfirmed, UC)	Есть один неподтвержденный источник или несколько конфликтующих отчетов	0,9
Не доказан (Uncorroborated, UR)	Есть несколько неофициальных источников, возможно, включая независимые компании по безопасности или исследовательские организации. На данном этапе могут конфликтовать технические детали	0,95
Подтвержден (Confirmed, C)	Уязвимость признана производителем или существование уязвимости подтверждено публикацией эксплоита	1,00
Не определено (Not defined, ND)	Назначение этого значения показателю позволит не изменять оценку уязвимости	1,00

В группу контекстных показателей входят: *потенциал побочного ущерба* (*Collateral damage potential*, *CDP*), *плотность целей* (*Target distribution*, *TD*), *требования безопасности* (*Security requirements*, *CR*, *IR*, *AR*). Среда может существенно повлиять на риск, который представляет уязвимость для организации. *Потенциал побочного ущерба* определяет потенциал потерь активов из-за разрушений или кражи свойств или оборудования. А также может измерять экономические потери производительности или доходов. Чем выше побочный ущерб, тем выше оценка уязвимости. *Плотность целей* определяет процент уязвимых систем. Чем выше пропорция уязвимых систем, тем выше оценка. *Требования безопасности* позволяют настраивать CVSS оценку в зависимости от важности актива для организации, в терминах конфиденциальности, целостности и доступности. Чем выше требование безопасности, тем выше оценка («среднее» считается значением по умолчанию).

Возможные значения контекстных показателей приведены в таблице 2.4.

Таблица 2.4

Значения, принимаемые контекстными показателями CVSS (версия 2.0)

Качественное значение показателя	Описание	Численное значение показателя
Потенциал побочного ущерба (Collateral damage potential)		
Нет (None, N)	Нет потенциала для потери активов, производительности или дохода	0
Низкий (Low, L)	Успешная эксплуатация уязвимости может привести к небольшому физическому разрушению, или имущественному ущербу. Или м.б. небольшая потеря дохода или продуктивности для организации	0,1

Качественное значение показателя	Описание	Численное значение показателя
Низкий-средний (Low-Medium, LM)	Успешная эксплуатация уязвимости может привести к среднему физическому или имущественному ущербу. Или м.б. средняя потеря дохода или продуктивности для организации	0,3
Средний-высокий (Medium-High, MH)	Успешная эксплуатация уязвимости может привести к серьезному физическому ущербу или потерям. Или м.б. серьезная потеря дохода или продуктивности для организации	0,4
Высокий (High, H)	Успешная эксплуатация уязвимости может привести к катастрофическому физическому или имущественному ущербу и потерям. Или м.б. катастрофическая потеря дохода или продуктивности для организации	0,5
Не определено (Not defined, ND)	Назначение этого значения показателю позволит не изменять оценку уязвимости	0
Плотность целей (Target distribution)		
Нет (None, N)	Нет целевых систем. 0% среды под риском	0
Низкое (Low, L)	Целевые узлы есть, но в небольшом количестве. 1–25% среды под риском	0,25
Среднее (Medium, M)	Целевые узлы есть, но среднее количество. 26–75% среды под риском	0,75
Высокое (High, H)	Целевые узлы есть в ощутимом количестве. 76–100%	1,00
Не определено (Not defined, ND)	Назначение этого значения показателю позволит не изменять оценку уязвимости	1,00
Требования безопасности (Security requirements)		
Низкие (Low, L)	Потеря конфиденциальности/целостности/доступности будет иметь минимальный неблагоприятный эффект на организацию	0,5
Средние (Medium, M)	Потеря конфиденциальности/целостности/доступности будет иметь серьезный неблагоприятный эффект на организацию	1,00
Высокие (High, H)	Потеря конфиденциальности/целостности/доступности будет иметь катастрофический неблагоприятный эффект на организацию	1,51
Не определено (Not defined, ND)	Назначение этого значения показателю позволит не изменять оценку уязвимости	1,0

Формат CVSS версии 2.0 имеет ряд неопределенностей в описании показателей и их возможных значений [99]. В 2015 году вышла новая версия CVSS [100], которая учитывает проблемы предыдущей версии. В 2019 г. вышло еще одно обновление — CVSS версии 3.1 [101].

Детальное сравнение метрик CVSS версии 2.0 и CVSS версии 3.0 приведено в таблице 2.5. Сравнение показало, что все метрики были изменены в той или иной степени. В таблице 2.5 выделены серым цветом знасительно измененные метрики/значения (например, вновь добавленные или удаленные). Остальные метрики подверглись небольшим изменениям (например, численных значений).

Таблица 2.5

Сравнение метрик CVSS версии 2.0 и 3.0

CVSS версии 2.0	CVSS версии 3.0
Группа метрик *Возможность использования* (*Exploitability*)	
***Вектор доступа* (*Access Vector*, *AV*)**	***Вектор атаки* (*Attack Vector*, *AV*)**
Значения *AV*	Значения *AV*
Локальный (Local, L): 0,395 — для эксплуатации уязвимости требуется локальный доступ к хосту	Локальный (Local, L): 0,55 — атакующему необходимы права на чтение/запись/запуск, чтобы эксплуатировать уязвимость. То есть атакующий должен либо быть залогинен в системе, либо положиться на взаимодействие с пользователем
	Физический (Physical, P): 0,2 — требует физических манипуляций с уязвимым компонентом
Смежная сеть (Adjacent Network, A): 0,646 — для эксплуатации уязвимости требуется доступ к смежной сети	Смежный (Adjacent, A): 0,62 — для эксплуатации уязвимости требуется доступ к смежной сети
Сетевой (Network, N): 1,0 — для эксплуатации уязвимости требуется сетевой доступ	Сетевой (Network, N): 0,85 — для эксплуатации уязвимости требуется сетевой доступ
***Сложность доступа* (*Access Complexity*, *AC*)**	***Сложность атаки* (*Attack Complexity*, *AC*)**
Значения *AC*	Значения *AC*
Высокий (High, H): 0,35 — высокая сложность эксплуатации уязвимости	Высокий (High, H): 0,44 — успех атаки зависит от условий вне контроля атакующего
Средний (Medium, M): 0,61 — средняя сложность эксплуатации уязвимости	
Низкий (Low, L): 0,71 — низкая сложность эксплуатации уязвимости	Низкий (Low, L): 0,77 — нет особых условий доступа. Атакующий может ожидать повторяемого успеха против уязвимого компонента
***Аутентификация* (*Authentication*, *Au*)**	***Требуемые привилегии* (*Privileges Required*, *PR*)**
Значения *Au*	Значения *PR*
Множественная (Multiple, M): 0,45 — для эксплуатации уязвимости требуется дополнительно пройти множество процедур аутентификации	Высокие (High, H): 0,27 (0,5, если изменилось значение области действия *Scope*) — атакующий авторизован и имеет привилегии, дающие значительный (административный) доступ к уязвимому компоненту, что может повлиять на настройки и файлы всей системы
Одиночная (Single, S): 0,56 — для эксплуатации уязвимости требуется дополнительно пройти одну процедуру аутентификации	Низкие (Low, L): 0,62 (0,68, если изменилось значение области действия *Scope*) — атакующий авторизован и имеет привилегии, дающие базовые пользовательские возможности, которые влияют только на файлы и настройки данного пользователя. Или может влиять только на не конфиденциальные ресурсы
Не требуется (None, N): 0,704 — для эксплуатации уязвимости дополнительной аутентификации не требуется	Не требуется (None, N): 0,85 — атакующий не авторизован, то есть доступа к настройкам и файлам не требуется

CVSS версии 2.0	**CVSS версии 3.0**
—	***User Interaction*, *UI***
	Значения *UI*
	Не требуется (None): 0,85 — система может быть скомпрометирована без участия пользователя
	Требуется (Required): 0,62 — пользователь должен совершить какие-то действия до того, как уязвимость будет проэксплуатирована. Например, успешный эксплоит возможен только в случае установки приложения системным администратором
Группа метрик *Влияние* (*Impact*)	
***Влияние на конфиденциальность* (*Confidentiality Impact*, *C*)**	***Влияние на конфиденциальность* (*Confidentiality Impact*, *C*)**
Значения *C*	Значения *C*
Нет (None, N): 0,0 — нет ущерба	Нет (None, N): 0,0
Частичный (Partial, P): 0,275 — частичный ущерб	Низкое (Low, L): 0,22
Полный (Complete, C): 0,660 — полный ущерб	Высокое (High, H): 0,56
***Влияние на целостность* (*Integrity Impact*, *I*)**	***Влияние на целостность* (*Integrity Impact*, *I*)**
Значения *I*	Значения *I*
Нет (None, N): 0,0	Нет (None, N): 0,0
Частичный (Partial, P): 0,275	Низкое (Low, L): 0,22
Полный (Complete, C): 0,660	Высокое (High, H): 0,56
***Влияние на доступность* (*Availability Impact*, *A*)**	***Влияние на доступность* (*Availability Impact*, *A*)**
Значения *A*	Значения *A*
Нет (None, N): 0,0	Нет (None, N): 0,0
Частичный (Partial, P): 0,275	Низкое (Low, L): 0,22
Полный (Complete, C): 0,660	Высокое (High, H): 0,56
—	***Область действия* (*Scope*, *S*)**
	Значения *S*
	Не меняется (Unchanged, U) — уязвимость влияет только на ресурсы в рамках привилегий уязвимого компонента: уязвимый компонент и подверженный влиянию компонент — один и тот же
	Меняется (Changed, C) — уязвимость влияет на ресурсы вне привилегий уязвимого компонента, в этом случае уязвимый компонент и подверженный влиянию компонент — разные

Оценки уязвимостей по CVSS определяются на основе уравнений ниже.

Вначале подсчитываются показатели базовой оценки *BaseScore* на основе базового уравнения:

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

где *Impact* — влияние на кибербезопасность объекта в случае успешного использования уязвимости;

Exploitability — возможность использования уязвимости;

$$f(Impact) = \begin{cases} 0, & \text{если } Impact = 0 \\ 0{,}176, & \text{если } Impact \neq 0 \end{cases};$$

функция round_to_1_dicimal выполняет округление аргумента до одного знака после запятой.

Влияние на кибербезопасность объекта *Impact* вычисляется по формуле:

$$Impact = 10{,}41 \times \left(1 - (1 - ConfImpact) \times (1 - IntegImpact) \times (1 - AvailImpact)\right),$$

где *ConfImpact*, *IntegImpact*, *AvailImpact* — влияние на конфиденциальность, целостность и доступность.

Возможность использования уязвимости *Exploitability* вычисляется по формуле:

$$Exploitability = 20 \times AccessVector \times AccessComplexity \times Authentication,$$

где *AccessVector — вектор доступа*;

AccessComplexity — сложность доступа;

Authentication — аутентификация.

Временное уравнение:

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

где *RemediationLevel — уровень исправления* уязвимости;

ReportConfidence — степень достоверности отчета об уязвимости.

Временная оценка не может быть выше, чем базовая, и не меньше, чем на 33% ниже нее.

Контекстное уравнение:

$$EnvironmentalScore = round\ to\ 1\ decimal((AdjustedTemporal + (10 - AdjustedTemporal) \times \times CollateralDamagePotential) \times TargetDistribution),$$

где $AdjustedTemporal = TemporalScore$, в котором *BaseScore Impact* заменен на $AdjustedImpact$;

$AdjustedImpact$ – влияние на кибербезопасность объекта в случае успешного использования уязвимости с учетом требований безопасности;

$CollateralDamagePotential$ – *потенциал побочного ущерба* при эксплуатации уязвимости;

$TargetDistribution$ — *плотность целей*.

Влияние на кибербезопасность объекта в случае успешного использования уязвимости с учетом требований безопасности вычисляется по формуле:

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

где $ConfReq$, $IntegReq$, $AvailReq$ — требования безопасности.

В связи с изменениями метрик CVSS и их значений формулы в CVSS 3.0 были также изменены. Формулы CVSS версии 2.0 и версии 3.0 приведены в таблице 2.6 для сравнения (обозначения метрик взяты из таблицы 2.5).

Кроме того, при определении оценки уязвимостей формируются векторы качественных значений показателей, обеспечивающие открытость оценки по CVSS (то есть раскрывающее, на основе чего она получена).

Спецификация каждого показателя вектора состоит из обозначения показателя, за которым следует знак «:» (двоеточие), а за ним — значения показателя. Показатели расположены внутри вектора в определенном порядке и разделены знаком «/» (слэш). Описания базового, временного и контекстного векторов приведены в таблице 2.7. В квадратных скобках показаны

возможные значения каждого показателя. Например, уязвимость со значениями базовых показателей «*вектор доступа* (*AC*): низкий (L), *сложность доступа* (*AC*): средняя (M), *аутентификация* (*Au*): нет (N), *влияние на конфиденциальность* (*C*): нет (N), *влияние на целостность* (*I*): частичное (P), *влияние на доступность* (*A*): полное (C)» будет иметь следующий базовый вектор: "*AV*:L/*AC*:M/*Au*:N/*C*:N/*I*:P/*A*:C".

Таблица 2.6

Формулы CVSS в версии 2.0 и версии 3.0

CVSS версии 2.0	CVSS версии 3.0
CVSS BaseScore= =round_to_1_decimal(((0,6×*Impact*)+ +(0,4×*Exploitability*) — 1,5)×*f*(*Impact*))	Если (*Impact* <= 0) *CVSS BaseScore*=0, иначе если (*Scope*=Unchanged) *CVSS BaseScore*=Roundup(min[(*Impact*+*Exploitability*), 10]), иначе если (*Scope*=Changed) *CVSS BaseScore*=Roundup(min[1,08×(*Impact*+*Exploitability*), 10])
Impact=10,41×(1-(1-*C*)×(1-*I*)×(1-*A*))	Если (*Scope*=Unchanged) *Impact* = 6,42×ISC_{Base}, иначе если (*Scope*=Changed) *Impact*=7,52×(ISC_{Base}-0,029) — 3,25×(ISC_{Base}-0,02)[15]
	ISC_{Base}=1-[(1-*C*)×(1-*I*)×(1-*A*)]
Exploitability=20×*AV*×*AC*×*Au*	*Exploitability*=8,22×*AV*×*AC*×*PR*×*UI*
Если (*Impact*=0) *f*(*Impact*)=0, иначе *f*(*Impact*)=1,176	—
round_to_1_decimal — функция округления до одного десятичного знака после запятой в большую сторону	Roundup — округление до одного десятичного знака после запятой в большую сторону

Таблица 2.7

Базовый, временной и контекстный векторы

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

На основе рассмотрения стандарта CVSS можно выделить следующие его преимущества:

— четкое определение показателей;

— стандартизованные оценки уязвимостей (это позволяет выделить уязвимости системы, которые требуют наиболее спешных мер), разрабатываемые группой профессионалов;

— открытость системы (то есть способ назначения оценок и свойства, определившие оценку, являются открытыми);

— возможность приоритизации рисков (контекстные оценки показывают, какой риск представляет уязвимость для конкретной системы).

Недостатки CVSS:

— уязвимости оцениваются независимо друг от друга.

— при оценивании уязвимости учитывается только прямой ущерб для целевого хоста.

Система оценки критичности уязвимостей Microsoft. Система оценки критичности уязвимостей Microsoft — Severity Rating System — относится к системам качественного ранжирования. Она учитывает при назначении оценок сложность использования уязвимости и общее воздействие на безопасность при ее использовании [102]. Возможные качественные значения критичности уязвимостей по Microsoft Severity Rating System представлены в таблице 2.8.

Таблица 2.8

Значения критичности уязвимостей по Microsoft Severity Rating System [102]

Значения критичности уязвимостей	Описание
Критичная	Использование уязвимости позволяет запустить код без взаимодействия с пользователем. Могут использоваться самораспространяющиеся вредоносные программы (например, сетевые черви) или ситуации, когда выполнение кода происходит без предупреждений, например, при просмотре веб-страницы или открытии электронной почты.
Важная	Использование уязвимости может привести к нарушению конфиденциальности, целостности или доступности данных пользователя, или нарушению конфиденциальности, целостности или доступности обрабатываемых ресурсов.
Средняя	Влияние уязвимости снижено такими факторами, как требования аутентификации или применимость только к нестандартным конфигурациям.
Низкая	Влияние уязвимости полностью снижено характеристиками уязвимого компонента.

Индекс возможности использования Microsoft. Кроме того, для оценки вероятности использования уязвимости Microsoft предлагается «Индекс возможности использования» (Exploitability Index). Этот показатель связан с обновлениями от Microsoft и характеризует вероятность использования уязвимостей, устраняемых обновлениями безопасности Microsoft, в течение тридцати дней после выпуска обновления [103]. Используется количественное ранжирование — показатель может принимать одно из четырех значений, приведенных в таблице 2.9. Недостатком данной системы является то, что, хотя используется количественное ранжирование, описания каждого уровня остаются вербальными.

Таблица 2.9

Значения показателя «Индекс возможности использования» от Microsoft [103]

Значения показателя «Индекс возможности использования»	Описание
0	Обнаружено использование.
1	Использование более вероятно: код эксплоита может быть создан таким образом, чтобы злоумышленник мог постоянно использовать уязвимость, и были зафиксированы случаи использования уязвимости.
2	Использование менее вероятно: хотя код эксплоита может быть создан, это связано с затруднениями для злоумышленника, например, требуется определенный уровень навыков или много времени, и/или результаты использования эксплоита могут быть разными.
3	Использование невероятно.

Система оценивания уязвимостей IP360. Система оценивания уязвимостей IP360 [104, 105] использует комплексный показатель для оценки уязвимостей:

$$V = \sqrt{t \times \frac{r!}{s^2}},$$

где t — количество дней, которые прошли с того момента, как информация об уязвимости впервые стала доступна;

r — фактор «класс риска», который отражает угрозу от наличия уязвимости. Это может быть шесть «классов риска» от 1 до 6 (классы приведены в таблице 2.10);

s — мера «набора навыков», необходимого для успешного проведения атаки, использующей данную уязвимость, определяется по шкале от 1 до 6 в зависимости от инструментов, используемых при эксплуатации уязвимостей (таблица 2.11).

Таблица 2.10

Значения фактора «класс риска» комплексного показателя для оценки уязвимостей IP360 [104, 105]

Значения фактора «класс риска»	Описание
1	Локальные атаки против доступности ресурса (например, различные локальные DoS атаки).
2	Локальные методы увеличения привилегий пользователя.
3	Локальные методы получения полных привилегий администратора.
4	Удаленные атаки против доступности ресурса.
5	Удаленные методы увеличения привилегий пользователя.
6	Удаленные методы получения привилегий администратора.

Таблица 2.11

Значения меры «набор навыков» комплексного показателя для оценки уязвимостей IP360 [104, 105]

Метка набора навыков	Описание требуемых инструментов	Значения меры «набор навыков» (s)	s^2
Автоматический эксплоит	Графическое приложение, которое включает установщик, либо эксплоит, который не требует взаимодействия с человеком, например, сетевой червь.	1	1
Простой	Не UNIX бинарное приложение, которое обычно включает установочный скрипт, пакетный файл или другой простой механизм установки. Бинарное приложение является скомпилированным эксплоитом, не требующим определенных знаний ОС или сети.	2	4
Средний	Не Windows бинарное приложение, которое обычно включает установочный скрипт, пакетный файл или другой простой механизм установки.	3	9
Сложный	Не Windows оболочка операционной системы, perl, или интерпретируемая программа скрипт, требующая ограниченных знаний ОС, кода оболочки операционной системы, интерпретаторов или сетей.	4	16

Метка набора навыков	Описание требуемых инструментов	Значения меры «набор навыков» (s)	s^2
Очень сложный	Нескомпилированный набор исходных файлов, обычно сжатых, требующий определенных знаний ОС, компиляторов и продвинутого системного опыта.	5	25
Неизвестный эксплоит	Обычно эта категория описывает эксплоит для которого не представлен исходный код, скрипт или источник.	6	36

Отметим основные недостатки и ограничения перечисленных методов оценки уязвимостей. Всем перечисленным методам количественного и качественного ранжирования в разной степени присущи следующие недостатки: субъективность, неоднозначность, неточность оценивания, нерелевантность контексту использования и в целом отсутствие доверия к оценке бизнес-рисков [104, 105].

Для формирования оценок бизнес-рисков, которым можно доверять, методы оценки уязвимостей должны быть объективными. Тем не менее методы количественного и качественного ранжирования не удовлетворяют этому требованию: ни категории, ни числа не содержат определения своей сущности.

Основными недостатками методов качественного ранжирования также является достаточно небольшое число уровней и невозможность их агрегирования. Использование чисел позволяет выделить существенно большее количество уровней. Однако другими существенными недостатками методов ранжирования уязвимостей является неоднозначность и неточность получаемых оценок, так как ранги позволяют выявлять только относительное различие между уязвимостями, и точность ранжирования снижается с ростом количества уязвимостей. Методы оценки на основе показателей в меньшей степени подвержены этим недостаткам, так как показатель представляет атомарное измерение уязвимости, не зависящее других существующих уязвимостей. Тем не менее оценки, определяемые на основе показателей, остаются субъективными.

Методы количественного и качественного ранжирования уязвимостей также подвержены недостатку нерелевантности жизненному циклу уязвимости и контексту применения. То есть выставленная оценка уязвимости не меняется со временем и не учитывает условий, присущих среде пользователя или организации. В системе на основе показателей IP360 учитывается изменение рисков во времени, но не учитывается контекст применения. Система CVSS не подвержена этому недостатку и учитывает изменение рисков во времени с помощью временной оценки, а контекст применения — с помощью контекстной оценки.

2.3. Общее перечисление слабых мест

Стандарт «Общее перечисление слабых мест» (Common Weakness Enumeration, CWE) представляет собой открытый словарь типов слабых мест программного обеспечения [106]. Словарь имеет несколько иерархических представлений (в виде графа): с точки зрения разработки ПО; с точ-

ки зрения проектирования аппаратного обеспечения; с точки зрения исследований. Схема CWE содержит описание основных полей CWE [107].

Схема включает элементы, свойства, типы и перечисления, которые описаны ниже. На рисунке 2.5 отображены элементы и свойства схемы первых двух уровней иерархии (свойства выделены зеленым цветом).

На верхнем уровне находится **элемент** схемы **каталог слабых мест** (Weakness_Catalog), который включает **элементы**: *слабые места* (Weaknesses); *категории* (Categories); *представления* (Views) и *внешние ссылки* (External_References). Кроме того, в него входят обязательные свойства: *имя* (Name), *версия* (Version), *дата* (Date).

Каталог слабых мест (Weakness_Catalog) является корневым элементом, используемым для описания проблем безопасности, известных как слабые места.

Элемент *слабые места* определяет слабые места CWE.

Элемент *категории* используется для группировки слабых мест, например, слабое место CWE-310, объединяющее криптографические проблемы. Элемент *представления* определяет различные перспективы, с которых можно смотреть на CWE, например, CWE-658 включает слабые места, найденные в языке C. Данные элементы определяют структуру каталога.

Элемент *внешние ссылки* (External_References) представляет собой ссылки, которые могут быть общими для отдельных слабых мест.

Последующие элементы иерархии зависят от типов предыдущих элементов. Для этого в схеме CWE выделены следующие **типы**: *слабое место* (WeaknessType); *категория* (CategoryType); *представление* (ViewType); *внешняя ссылка* (ExternalReferenceType).

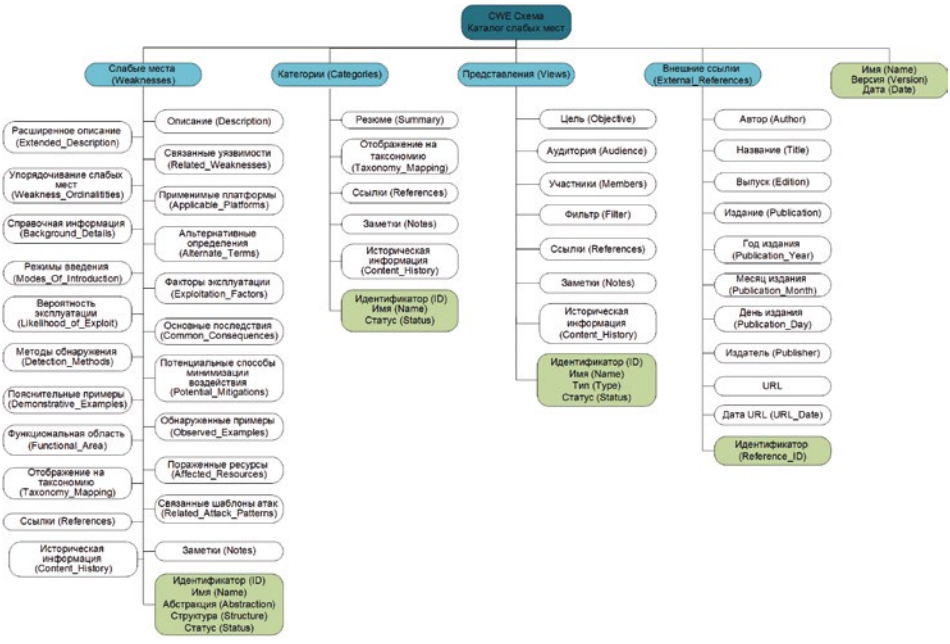


Рис. 2.5. Основные элементы схемы CWE

Элементы типа *слабое место* (WeaknessType) включают следующие вложенные элементы: *описание* (Description, обязательный элемент); расширенное *описание* (Extended_Description); *связанные уязвимости* (Related_Weaknesses); *упорядочивание слабых мест* (Weakness_Ordinalitities); *применимые платформы* (Applicable_Platforms); *справочная информация* (Background_Details); *альтернативные определения* (Alternate_Terms); *режимы введения* (Modes_Of_Introduction); *факторы эксплуатации* (Exploitation_Factors); *вероятность эксплуатации* (Likelihood_of_Exploit), которая может принимать значения из перечисления *вероятности* (LikelihoodEnumeration); *основные последствия* (Common_Consequences); *методы обнаружения* (Detection_Methods); *потенциальные способы минимизации воздействия* (Potential_Mitigations); *пояснительные примеры* (Demonstrative_Examples); *обнаруженные примеры* (Observed_Examples); *функциональная область* (Functional_Area); *пораженные ресурсы* (Affected_Resources); *отображение на таксономию* (Taxonomy_Mapping); *связанные шаблоны атак* (Related_Attack_Patterns); *ссылки* (References); *заметки* (Notes); *историческая информация* (Content_History). Кроме того, в него входят обязательные свойства: *идентификатор* (ID); *имя* (Name); *абстракция* (Abstraction), которое определяет уровень абстракции описания слабого места и может принимать значения из перечисления *абстракции* (AbstractionEnumeration); *структура* (Structure), которое определяет структурную природу слабого места и может принимать значения из перечисления *структуры* (StructureEnumeration), и *статус* (Status), которое определяет завершенность описания слабого места и может принимать значения из перечисления *статусы* (StatusEnumeration).

Элементы типа *категория* (CategoryType) включают следующие элементы: резюме (Summary); *отображение на таксономию* (Taxonomy_Mapping); *ссылки* (References); *заметки* (Notes); *историческая информация* (Content_History). Кроме того, в него входят обязательные свойства: *идентификатор* (ID); *имя* (Name); и *статус* (Status), которое определяет завершенность описания категории и может принимать значения из перечисления *статусы* (StatusEnumeration).

Элементы типа *представление* (ViewType) включают следующие элементы: *цель* (Objective, обязательный элемент); *аудитория* (Audience); *участники* (Members); *фильтр* (Filter); *ссылки* (References); *заметки* (Notes); *историческая информация* (Content_History). Кроме того, в него входят обязательные свойства: *идентификатор* (ID); *имя* (Name); *тип* (Type), которое определяет, каким образом заданный вид был сформирован и может принимать значения из перечисления *типы представления* (ViewTypeEnumeration), и *статус* (Status), которое определяет завершенность описания категории и может принимать значения из перечисления *статусы* (StatusEnumeration).

Элементы типа *внешняя ссылка* (ExternalReferenceType) включают следующие элементы: *автор* (Author); *название* (Title, обязательный элемент); *выпуск* (Edition); *издание* (Publication); *год издания* (Publication_Year); *месяц издания* (Publication_Month); *день издания* (Publication_Day); *издатель* (Publisher); URL; *дата URL* (URL_Date). Кроме того, в него входят обязательные свойства: *идентификатор* (Reference_ID).

Последующие элементы иерархии зависят от типов предыдущих элементов. Для этого в схеме CWE выделены следующие **типы**: *связанные уязвимости* (RelatedWeaknessesType); *упорядочивание слабых мест* (WeaknessOrdinalitiesType); *применимые платформы* (ApplicablePlatformsType); *справочная информация* (BackgroundDetailsType); *альтернативные определения* (AlternateTermsType); *режимы введения* (ModesOfIntroductionType); *факторы эксплуатации* (ExploitationFactorsType); *основные последствия* (CommonConsequencesType); *методы обнаружения* (DetectionMethodsType); *потенциальные способы минимизации воздействия* (PotentialMitigationsType); *пояснительные примеры* (DemonstrativeExamplesType); *обнаруженные примеры* (ObservedExampleType); *функциональная область* (FunctionalAreasType); *пораженные ресурсы* (AffectedResourcesType); *отображение на таксономию* (TaxonomyMappingsType); *связанные шаблоны атак* (RelatedAttackPatternsType); *ссылки* (ReferencesType); *заметки* (NotesType); *историческая информация* (ContentHistoryType); *аудитория* (AudienceType); *участники* (MemberType).

Элементы типа *связанные уязвимости* (RelatedWeaknessesType) включают следующие обязательные свойства: *природа* (Nature), описывает природу отношений и принимает значения из перечисления *природа* (RelatedNatureEnumeration); CWE_ID (идентификатор связанного слабого места); View_ID (идентификатор связанного представления). Включают следующие свойства: Chain_ID, уникальный идентификатор именованной цепочки, к которой относится отношение CanFollow (может следовать) или CanPrecede (может предшествовать); *порядок* (Ordinal), может принимать значения из перечисления *порядки* (OrdinalEnumeration).

Тип *упорядочивание слабых мест* (WeaknessOrdinalitiesType) определяет потенциальные упорядочивающие отношения с другими слабыми местами. Элементы данного типа включают обязательный элемент *упорядочивание* (Ordinality), который может принимать значения из перечисления *упорядочивание* (OrdinalityEnumeration), и элемент *описание* (Description), который описывает контекст, при котором отношение существует.

Тип *применимые платформы* (Applicable_Platforms) определяет языки, ОС, архитектуры и технологии, в которых может появиться слабое место. Элементы данного типа включают элементы: *язык* (Language) со свойствами *имя* (Name), которое может принимать значения из перечисления *названия языка* (LanguageNameEnumeration), *класс* (Class), которое может принимать значения из перечисления *классы языков* (LanguageClassEnumeration), и обязательным свойством *распространенность* (Prevalence), которое может принимать значения из перечисления *виды распространенности* (PrevalenceEnumeration); *операционная система* (Operating_System) со свойствами *имя* (Name), которое может принимать значения из перечисления *операционные системы* (OperatingSystemNameEnumeration), *версия* (Version), CPE_ID, *класс* (Class), которое может принимать значения из перечисления *классы операционных систем* (OperatingSystemClassEnumeration), и обязательным свойством *распространенность* (Prevalence); *архитектура* (Architecture) со свойствами *имя* (Name), которое может принимать значе-

ния из перечисления *названия архитектур* (ArchitectureNameEnumeration), *класс* (Class), которое может принимать значения из перечисления *классы архитектур* (ArchitectureClassEnumeration), и обязательным свойством *распространенность* (Prevalence), которое может принимать значения из перечисления *виды распространенности* (PrevalenceEnumeration); *технология* (Technology) со свойствами *имя* (Name), которое может принимать значения из перечисления *названия технологий* (TechnologyNameEnumeration), *класс* (Class), которое может принимать значения из перечисления *классы технологий* (TechnologyClassEnumeration), и обязательным свойством *распространенность* (Prevalence), которое может принимать значения из перечисления *виды распространенности* (PrevalenceEnumeration).

Элементы типа *справочная информация* (BackgroundDetailsType) содержат актуальную информацию, не относящуюся к природе слабого места, и включают один или более элементов *справочная информация* (Background_Detail).

Элементы типа *альтернативные определения* (AlternateTermsType) определяют одно или более других названий слабого места. Элементы такого типа включают обязательные элементы *термин* (Term) и *описание* (Description).

Тип *режимы введения* (ModesOfIntroductionType) позволяет определить, как и когда слабое место может быть внесено. Элементы данного типа включают обязательный элемент *введение* (Introduction), который, в свою очередь, включает обязательный элемент *фаза* (Phase), определяющий этап жизненного цикла продукта, на котором может появиться слабое место, и принимающий значения из перечисления *фазы* (PhaseEnumeration), и элемент *заметка* (Note), описывающий сценарий, при котором слабое место может появиться.

Тип *факторы эксплуатации* (ExploitationFactorsType) определяет условия или факторы, которые могут повысить вероятность использования слабого места.

Элементы типа *основные последствия* (CommonConsequencesType) включают обязательный элемент *область действия* (Scope), который определяет нарушаемое свойство безопасности и принимает значения из перечисления *области действия* (ScopeEnumeration), и элементы *ущерб* (Impact), описывающий технический ущерб в случае успешного использования слабого места атакующим и принимающий значения из перечисления *технический ущерб* (TechnicalImpactEnumeration), *вероятность* (Likelihood), определяющий насколько вероятно определенное последствие относительно других последствий и принимающий значения из перечисления *вероятности* (LikelihoodEnumeration), и *примечание* (Note). Кроме того, элементы типа *основные последствия* (CommonConsequencesType) включают свойство Consequence_ID (идентификатор последствия).

Тип *методы обнаружения* (DetectionMethodsType) позволяет определить методы, которые могут использоваться для обнаружения слабого места, их достоинства и ограничения. Элементы данного типа включают следующие обязательные элементы: *метод* (Method), принимает значения из перечисления *методы обнаружения* (DetectionMethodEnumeration);

описание (Description). Включают следующие элементы: *эффективность* (Effectiveness), принимает значения из перечисления *эффективность* (Det ectionEffectivenessEnumeration); *заметки* (Effectiveness_Notes). Кроме того, элементы типа *методы обнаружения* (DetectionMethodsType) включают свойство Detection_Method_ID (уникальный идентификатор метода).

Элементы типа *потенциальные способы минимизации воздействия* (PotentialMitigationsType) включают следующие элементы: *фаза* (Phase), указывает на каком этапе цикла разработки способ минимизации воздействия может быть применен и принимает значения из перечисления *фазы* (PhaseEnumeration); *стратегия* (Strategy), описывает стратегию защиты к которой относится способ минимизации воздействия и принимает значения из перечисления *стратегии минимизации воздействия* (Mitigatio nStrategyEnumeration); эффективность (Effectiveness), принимает значения из перечисления *эффективность* (EffectivenessEnumeration); *заметки* (Effectiveness_Notes). Включают следующие обязательные элементы: *описание* (Description). Кроме того, элементы данного типа включают свойство Mitigation_ID (уникальный идентификатор способа минимизации воздействия), имеющее формат MIT-1.

Тип *пояснительные примеры* (DemonstrativeExamplesType) позволяет определить примеры того, как выглядит слабое место в коде. Элементы данного типа включают следующие элементы: *заголовок* (Title_Text); *введение* (Intro_Text, обязательный элемент), описывающее контекст и настройки при которых следует просматривать код, и для чего он предназначен; *текст* (Body_Text), описывающий пример; *пример кода* (Example_Code); *ссылки* (References). Кроме того, элементы данного типа включают свойство Demonstrative_Example_ID (уникальный идентификатор примера).

Тип *обнаруженные примеры* (ObservedExampleType) дает возможность определить ссылки на наблюдаемые примеры слабых мест в реальных продуктах. Обычно это ссылка на уязвимость CVE. Элементы данного типа включают следующие обязательные элементы: *ссылка* (Reference); *описание* (Description); *ссылка на сайт* (Link), содержит URL с более подробной информацией о примере.

Тип *функциональная область* (FunctionalAreasType) позволяет определить функциональные области, в которых наиболее вероятно появление слабого места. Например, слабое место CWE-23 (обход относительного пути) может появиться в функциональных областях ПО, связанного с обработкой файлов. Элементы данного типа принимают значения из перечисления *функциональные области* (FunctionalAreaEnumeration).

Тип *пораженные ресурсы* (AffectedResourcesType) дает возможность определить ресурсы, которые могут быть поражены при эксплуатации слабого места. Элементы данного типа могут принимать значения из перечисления *ресурсы* (ResourceEnumeration).

Тип *отображение на таксономию* (TaxonomyMappingsType) используется для отображения записи (слабого места или категории) в CWE на эквивалентную запись в другой таксономии. Элементы данного типа включают следующие элементы: Entry_ID (идентификатор отображаемой записи); *имя*

записи (Entry_Name); *соответствие отображения* (Mapping_Fit), определяет, насколько близка CWE запись к записи в таксономии, и принимает значения из перечисления *соответствия отображения* (TaxonomyMappin gFitEnumeration). Кроме того, элементы типа *отображение на таксономию* (TaxonomyMappingsType) включают обязательное свойство *имя таксономии* (Taxonomy_Name), которое может принимать значения из перечисления *имена таксономий* (TaxonomyNameEnumeration).

Элементы типа *связанные шаблоны атак* (RelatedAttackPatternsType) включают обязательное свойство CAPEC_ID (идентификатор связанного шаблона атаки).

Элементы типа *заметки* (NotesType) содержат дополнительные комментарии о сущности, которые не вошли в другие элементы. Элементы данного типа включают обязательное свойство *тип* (Type), которое может принимать значения из перечисления *типы заметок* (NoteTypeEnumeration).

Тип *ссылки* (ReferencesType) используется для связи с внешней ссылкой, определенной в каталоге. Элементы данного типа включают обязательное свойство External_Reference_ID (идентификатор внешней ссылки) в формате REF-1 и свойство *раздел* (Section), содержащее название раздела или номер страницы по ссылке.

Тип *историческая информация* (ContentHistoryType) позволяет сохранить историю изменений записи о слабом месте, начиная с ее автора. Элементы данного типа включают следующие элементы: *подача* (Submission, обязательный элемент), который, в свою очередь, включает элементы *название* (Submission_Name), *организация* (Submission_Organization), *дата подачи* (Submission_Date) и *комментарий* (Submission_Comment); *изменение* (Modification), который, в свою очередь, включает элементы *имя* (Modification_Name), *организация* (Modification_Organization), *дата изменения* (Modification_Date), *важность изменения* (Modification_ Importance), который может принимать значения из перечисления *важность* (ImportanceEnumeration) и *комментарий* (Modification_Comment); *вклад* (Contribution), который, в свою очередь, включает элементы *имя* (Contribution_Name), *организация* (Contribution_Organization), *дата* (Contribution_Date) и *комментарий* (Contribution_Comment), а также обязательное свойство *тип* (Type), которое указывает, был ли вклад частью обратной связи или пожертвованного фактического содержания; *предыдущее имя записи* (Previous_Entry_Name), который включает обязательное свойство *дата* (Date).

Тип *аудитория* (AudienceType) определяет заинтересованные стороны или группы, для которых предназначено *представление* (View). Элементы данного типа включают элемент *заинтересованная сторона* (Stakeholder, обязательный элемент), который, в свою очередь, включает обязательный элемент *тип* (Type) и элемент *описание* (Description).

Тип *участники* (MemberType) может использоваться для создания отношений типа *имеет участника* (Has_Member) или *участник* (MemberOf) между видом, категориями и слабыми местами внутри *вида*. Элементы данного типа имеют обязательные свойства CWE_ID (идентификатор слабого места) и View_ID (идентификатор представления).

Примеры **перечислений**, используемых в схеме CWE, приведены в таблице 2.12. Полный список перечислений приведен в приложении А (таблица А.1).

2.4. Общее перечисление и классификация шаблонов атак

Стандарт «Общее перечисление и классификация шаблонов атак» (Common Attack Pattern Enumeration and Classification, CAPEC) был создан корпорацией MITRE [94] для Министерства национальной безопасности США. Он включает в себя открытый каталог шаблонов атак (версия 3.4) и схему (версия 3.4) [10, 11, 108].

В стандарте CAPEC описываются не отдельные уязвимости и слабые места, а подходы и методики, используемые атакующими для компрометации компьютерных систем. CAPEC поддерживает три способа представления атак (для определения представления в схеме CAPEC выделен элемент *представление* (View)): иерархическое представление (граф), представление по связям с внешними факторами (внешний срез или внешние отображения), представление по связям с определенными атрибутами (внутренний срез или полезные представления).

CAPEC включает иерархическое представление по механизмам атак и по областям атак.

Верхний уровень представления по механизмам атак содержит девять основных категорий атак (для определения категорий в схеме CAPEC выделен элемент *категория* (Category)):

- вовлечение в мошеннические взаимодействия (engage in deceptive interactions);
- злоупотребление существующей функциональностью (abuse existing functionality);
- манипулирование структурами данных (manipulate data structures);
- манипулирование системными ресурсами (manipulate system resources);
- внедрение неожиданных пунктов (inject unexpected items);
- применение вероятностных методик (employ probabilistic techniques);
- манипулирование временем и состоянием (manipulate timing and state);
- сбор и анализ информации (collect and analyze information);
- разрушение контроля доступа (subvert access control).

Таблица 2.12

Примеры перечислений, используемых в схеме CWE [107]

Название перечисления	Возможные значения	Описание значений
Вероятности (LikelihoodEnumeration)	Высокая (High)	
	Средняя (Medium)	
	Низкая (Low)	
	Неизвестная (Unknown)	
Уровни абстракции (AbstractionEnumeration)	Столп (Pillar)	Наивысший уровень абстракции, отображает тему всех слабых мест с уровнем абстракции «класс» (class), «основа» (base), «вариант» (variant).

Название перечисления	Возможные значения	Описание значений
Уровни абстракции (AbstractionEnumeration)	Класс (Class)	Слабое место описывается абстрактно (абстрактнее, чем в случае абстракции Base), независимо от языка и технологии, но более конкретно, чем в случае уровня абстракции Pillar. Обычно для описания слабого места используется 1 или 2 из следующих аспектов: поведение, свойство, ресурс.
	Основа (Base)	Более конкретный тип слабого места, в большинстве случаев независимый от языка и технологии, но с достаточной детализацией, позволяющей определить методы обнаружения и предотвращения применения. Обычно для описания слабого места используется от 2 до 3 из следующих аспектов: поведение, свойство, технология, язык, ресурс.
	Вариант (Variant)	Слабое место связано с определенным типом продукта, конкретным языком и технологией. Описывается более конкретно, чем в случае уровня абстракции Base. Обычно для описания слабого места используется от 3 до 5 из следующих аспектов: поведение, свойство, технология, язык, ресурс.
	Составной (Compound)	Значимое объединение нескольких слабых мест.
Структуры (StructureEnumeration)	Цепочка (Chain)	Набор слабых мест, которые должны быть доступны последовательно, чтобы возникла эксплуатируемая уязвимость.
	Составной (Composite)	Набор слабых мест, которые должны одновременно присутствовать, чтобы возникла эксплуатируемая уязвимость.
	Простой (Simple)	Отдельное слабое место, использование которого не зависит от присутствия другого слабого места.
Статусы (StatusEnumera-tion)	Устарела (Deprecated)	Запись, удаленная из CWE, т.к. является дубликатом или была создана по ошибке.
	Черновик (Draft)	Все важные элементы записи заполнены и критичные элементы, такие как *имя* и *описание* хорошо описаны.
	Не завершена (Incomplete)	Не все важные элементы записи заполнены и нет гарантии качества.
	Вышла из употребления (Obsolete)	Запись корректна, но не является актуальной, например, ввиду замены записью новее.
	Стабильная (Stable)	Все важные элементы верифицированы и запись, скорее всего, не изменится в будущем.
	Употребимая (Usable)	Проведена полноценная проверка, и верифицированы критичные элементы.

Верхний уровень представления по областям атак содержит шесть основных категорий атак:

- социальная инженерия (social engineering);
- цепи поставок (supply chain);
- средства связи (communications);
- программное обеспечение (software);
- физическая безопасность (physical security);
- аппаратное обеспечение (hardware).

Каждая категория содержит структурированные данные о шаблонах атак (для определения шаблона атаки в схеме CAPEC выделен элемент *шаблон атаки* (Attack_Pattern)), включая описание, процесс реализации атаки, критичность (severity), примеры, связи с элементами CWE [9] и другие поля.

Представление по связям с внешними факторами (внешние отображения) включает шаблоны атак, объединенные внешним фактором: представление по классификации угроз консорциума по безопасности веб-приложений (Web Application Security Consortium, WASC) — WASC Threat Classification 2.0 [109]; представление по шаблонам атак ATT&CK [14]; представление по шаблонам атак OWASP [110].

Представление по связям с определенными атрибутами (полезные представления) объединяет шаблоны атак по определенному атрибуту из списка ниже:

- мобильные устройства;
- полный словарь CAPEC;
- метаабстракции;
- стандартные абстракции;
- детальные абстракции;
- устаревшие записи.

Более подробное описание схемы CAPEC для отображения шаблонов атак представлено ниже [108, 111].

В [111] выделяются основные и вспомогательные элементы схемы CAPEC. Эти элементы представлены на рисунке 2.6 и рисунке 2.7 соответственно.

Основные элементы схемы CAPEC.

Основные элементы схемы CAPEC включают следующие типы информации: идентификационная информация (identifying information), описательная информация (describing information), предписывающая информация (prescribing information), обзорная и разграничивающая информация (scoping and delimiting information), административная информация (administrative information).

Идентификационная информация включает свойства *идентификатор представления/категории/шаблона атаки* (ID), *идентификатор внешней ссылки* (Reference_ID) и *имя представления/категории/шаблона атаки* (Name).

Представление (View) определяет способ отображения шаблонов атак CAPEC. Как указано выше, в CAPEC выделяется три способа представления: граф, внешний срез и внутренний срез.

Категория (Category) представляет собой набор шаблонов атак, выделенный на основе общей характеристики.

Шаблон атаки (Attack_Pattern) представляет собой абстрактное описание того, как выполняется атака.

Идентификатор представления/категории/шаблона атаки — это уникальный идентификатор целого типа. Форма идентификатора категории/шаблона атаки для доступа извне: “CAPEC-####” (например, CAPEC-12). *Имя представления/категории/шаблона атаки* — это короткое описательное имя представления/категории/шаблона.

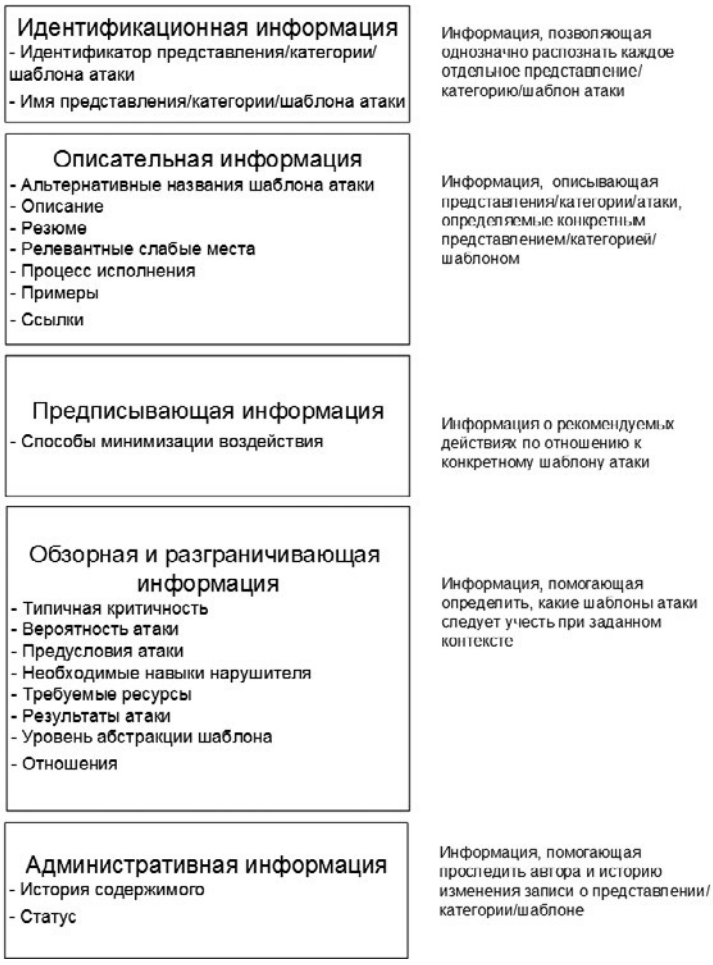


Рис. 2.6. Основные элементы схемы CAPEC

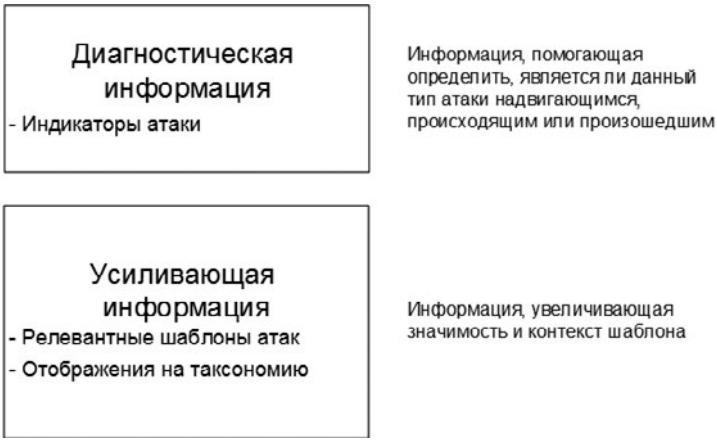


Рис. 2.7. Вспомогательные элементы схемы CAPEC

Внешняя ссылка (External_Reference) представляет собой указатель на расположение более детальной информации.

Описательная информация в случае представления включает элементы: *ссылки* (References). В случае категории описательная информация включает элементы: *резюме* (Summary) и *ссылки* (References). В случае шаблона атаки описательная информация включает элементы: *альтернативные названия* (Alternate_Terms), *описание* (Description), *релевантные слабые места* (Related_Weaknesses), *примеры* (Examples-Instances), *ссылки* (References), *процесс исполнения* (Execution_Flow).

Ссылки (References) — это перечисление ресурсов, которые использовались для разработки определения данного представления/категории/шаблона атаки, и ресурсов, которые могут показать ценность поиска дальнейшей информации об атаке.

Резюме (Summary) задает краткое описание категории.

Описание (Description) — это верхнеуровневое описание атаки, которое включает описание того, как поставляются злонамеренные входные данные, какое используется слабое место и итоговый технический ущерб.

Релевантные слабые места (Related_Weaknesses) — это информация о том, на какие слабые места нацелена данная атака. Поле ссылается на стандарт CWE [107]. Каждое релевантное слабое место описывается *идентификатором CWE* (CWE_ID).

Процесс исполнения (Execution_Flow) *атаки* описывает шаги, которые выполняет атакующий при типичной реализации атаки. Каждый шаг относится к определенной *фазе атаки* (Phase). *Фаза атаки* может принимать следующие значения: «разведка» (Explore), «эксперимент» (Experiment) или «действие» (Exploit). Каждый шаг атаки — это краткое описание конкретного шага атаки, задаваемое с помощью следующей информации: *номер шага атаки* (Step), *описание шага атаки* (Description), и *методики шага атаки* (Techniques), которая имеет *свойство* CAPEC_ID. Подробную информацию по этим полям можно найти в [108].

Примеры (Examples-Instances) — это примеры использования конкретной атаки.

Предписывающая информация включает поля, предоставляющие информацию о рекомендуемых действиях по отношению к шаблону атаки: *способы минимизации воздействия* (Mitigations). Поле *способы минимизации воздействия* определяет действия и подходы, которые могут предотвратить или снизить риск рассматриваемого типа атаки.

Обзорная и разграничивающая информация помогает определить, какие шаблоны атаки актуальны в данном контексте. Она включает поля: *типичная критичность* (Typical_Severity)*, вероятность атаки* (Likelihood_of_Attack), *предусловия атаки* (Prerequisites), *необходимые навыки нарушителя* (Skills_Required), *требуемые ресурсы* (Resources_Required), *результаты атаки* (Consequences), *уровень абстракции шаблона* (Abstraction), *отношения* (Relationships).

Типичная критичность показывает, какова критичность воздействия на целевое ПО в случае успешной атаки. Она принимает значения на шкале «очень низкая, низкая, средняя, высокая, очень высокая».

Вероятность атаки (Likelihood_of_Attack) показывает, какова общая вероятность успеха конкретного типа атаки. Она принимает значения на шкале «неизвестно, низкая, средняя, высокая».

Предусловия атаки (Prerequisites) — это условия, которые должны соблюдаться или функциональность и характеристики целевого ПО (или его поведения), необходимые для успеха атаки данного типа.

Необходимые навыки нарушителя (Skills_Required) — это уровень навыков или определенных знаний, необходимый для проведения нарушителем данной атаки. Он может определяться на основе шкалы «неизвестно, низкий, средний, высокий».

Требуемые ресурсы (Resources_Required) — это ресурсы (циклы CPU, IP-адреса, инструменты и т.п.), необходимые атакующему для эффективного осуществления определенного типа атаки.

Результаты атаки (Consequences) — это то, чего пытается достигнуть нарушитель, используя атаку. Данный элемент описывается элементами: *область действия* (Scope), ущерб (Impact), *вероятность* (Likelihood) и *заметки* (Note). *Область действия* (Scope) может принимать значения из списка «конфиденциальность (Confidentiality), целостность (Integrity), доступность (Availability), контроль доступа (Access Control), подотчетность (Accountability), аутентификация (Authentication), авторизация (Authorization), неотказуемость (Non-Repudiation), другое (Other)».

Уровень абстракции шаблона — отражает соответствующий уровень абстракции шаблона («мета», «стандартный» или «детальный»), который помогает управлять информацией, необходимой для его определения, и то, где и как шаблон наиболее полезно используется.

Основные отличия шаблонов атаки метаабстракции, стандартной абстракции и детальной абстракции приведены в таблице 2.13 [111].

Отношения (Relationships) используются для определения отношения с шаблонами атак, категориями и представлениями.

Таблица 2.13

Отличия шаблонов атаки на различных уровнях абстракции

Характеристика / уровень абстракции шаблона	Мета	Стандартный	Детальный
Контекст	— технологический и функциональный контекст — не специфичный; — часто определяет подходы к обдумыванию того, как разрушить допущения, сделанные в ПО	— обычно функциональный контекст — специфичный, а технологический контекст — не специфичный; — часто определяет действенный подход к использованию определенного слабого места в ПО	— функциональный и/или технологический контекст — специфичный; — часто определяет определенный вектор доставки и/или содержание атаки

Характеристика / уровень абстракции шаблона	Мета	Стандартный	Детальный
Релевантные ресурсы знаний	— общие принципы защиты; — общие слабые места	— общие слабые места; — шаблоны проектирования; — шаблоны безопасности; — стандарты безопасного кодирования; — руководящие документы по безопасности	— общие слабые места; — общие уязвимости; — стандарты безопасного кодирования
Процессы, для которых шаблоны атак являются практически значимыми	написание высокоуровневой политики безопасности	— написание детальной политики безопасности; — требования безопасности (руководство для создания случаев злоупотребления); — разработка сценариев тестирования безопасности; — тестирование приложений проникновением	— требования безопасности (шаблоны для создания случаев злоупотребления); — анализ безопасности кода (косвенно через целевые слабые места); — разработка сценариев тестирования безопасности; — тестирование приложений проникновением
Решения и контрмеры	обычно образовательные или тривиальные	специфичный как по проектированию, так и по реализации	специфичный по реализации и в определенных случаях — по проектированию
Общая значимость	поддерживает анализ безопасности ПО и высокоуровневые заключения	поддерживает анализ безопасности ПО	поддерживают четкое определение и/или автоматизацию атаки

Административная информация включает на верхнем уровне элемент представления/категории/шаблона атаки *история содержимого* (Content_History) и свойство *статус* (Status), которое может принимать значения «устарел (Deprecated), черновик (Draft), не завершен (Incomplete), вышел из употребления (Obsolete), стабильный (Stable), употребимый (Usable)». Элемент *история содержимого* (Content_History) позволяет сохранить историю изменений записи о представлении/категории/шаблоне атаки начиная с ее автора. Элементы, описывающие историю содержимого записи: *представление информации* (Submission), включая *имя представления* (Submission_Name), *организацию представителя* (Submission_Organization), *дату представления* (Submission_Date) и *комментарий представления* (Submission_Comment); *изменение записи* (Modification), включая *имя изменения* (Modification_Name), *организацию изменения* (Modification_Organization), *дату изменения* (Modification_Date), *важность изменения* (Modification_Importance) и *комментарий изменения* (Modification Comment); *вклад* (Contribution), включая элементы *имя* (Contribution_Name),

организацию (Contribution_Organization), *дату* (Contribution_Date), *комментарий* (Contribution_Comment), а также обязательное свойство *тип* (Type), которое указывает, был ли вклад частью обратной связи или пожертвованного фактического содержания; *предыдущее имя записи* (Previous_Entry_Name), который включает обязательное свойство *дата* (Date).

Вспомогательные элементы схемы CAPEC. Вспомогательные элементы схемы CAPEC (рисунок 2.7) [111] делятся на две группы: диагностическая информация (Diagnosing Information) и усиливающая информация (Enhancing Information).

Диагностическая информация включает поля, содержащие информацию, которая помогает определить, является ли данный тип атаки потенциальным, происходящим или произошедшим. Это *индикаторы атаки* (Indicators).

Индикаторы атаки — это действия, события, условия или поведение, которое может служить индикатором того, что атака заданного типа произойдет, происходит или произошла.

Усиливающая информация включает поля, предназначенные для предоставления информации, усиливающей значение и контекст шаблона атаки. Например, это могут быть важные ссылки на другие каталоги знаний.

Рассмотрим подробнее соответствующие элементы: *релевантные шаблоны атак* (Related_Attack_Patterns) и *отображение на таксономию* (Taxonomy_Mappings).

Релевантные шаблоны атак (Related_Attack_Patterns) — это шаблоны, которые связаны или зависимы от заданного шаблона. Релевантные шаблоны описываются на основе следующей информации: *идентификатор релевантного шаблона атаки* (CAPEC ID) и *природа* (Nature). Элемент *природа* (Nature) описывает природу связи с шаблоном атак, он может принимать значения из списка «потомок (ChildOf), предок (ParentOf), начинается с (StartsWith), может следовать за (CanFollow), может предшествовать (CanPrecede), требуется (RequiredBy), требует (Requires), может также быть (CanAlsoBe), пара (PeerOf)».

Отображение на таксономию (Taxonomy_Mappings) — этот элемент позволяет задать связи категорий и шаблонов атак с записями в других таксономиях.

Примеры шаблонов атак из каталога CAPEC.

Рассмотрим несколько примеров шаблонов атак из каталога CAPEC.

В таблице 2.14 представлена категория атаки «инъекция неожиданных элементов» — “Inject Unexpected Items” [112]. Категория имеет идентификатор “CAPEC-152”. Для данной категории атак характерно то, что целевое приложение интерпретирует входные данные, в которые нарушитель включил свои инструкции. Это приводит к непредусмотренным в приложении действиям или нестабильности приложения.

Таблица 2.14

Категория атаки «инъекция неожиданных элементов»

Идентификатор	**152**
Имя категории	Инъекция неожиданных элементов (Inject Unexpected Items)
Статус (Status)	Стабильная
Описание (Description)	Нарушитель может контролировать или нарушать поведение целевого узла на основе ложных входных данных, введенных через интерфейс обработки входных данных, или путем установки и исполнения вредоносного кода в целевой системе. Первое происходит, если нарушитель добавляет на вход данные, которые интерпретируются приложением так, что оно выполняет шаги, не предусмотренные приложением, или приводят приложение в нестабильное состояние. Атаки этого типа отличаются от атак на Структуру Данных тем, что последние разрушают базовые структуры, которые содержат данные, предоставленные пользователем, либо упреждая интерпретацию ввода (в случае переполнения буфера), либо приводя к значениям, которые целевое приложение не может правильно обработать (в случае целочисленного переполнения). В атаках с использованием инъекций входные данные интерпретируются приложением, но нарушитель включил инструкции в функции интерпретации, которым затем следует целевое приложение.
Отношения (Relationships) с участниками (Member) и связанными слабыми местами (Related_ Weaknesses)	Участник (MemberOf) представления View-1000 (Механизмы атаки, Mechanisms of attack).
	Имеет участников (HasMember): меташаблоны атак CAPEC-137, CAPEC-175, CAPEC-240, CAPEC-242, CAPEC-248, CAPEC-549, CAPEC-86, CAPEC-594, CAPEC-624.
История содержимого (Content_History)	Представление информации (Submissions): Дата представления (Submission Date): 2014-06-23; Представитель (Submitter): CAPEC Content Team; Организация (Organization): The MITRE Corporation.
	Изменения (Modifications): Дата изменения (Modification Date): 2015-11-09; Внес изменения (Modifier): CAPEC Content Team, обновлены отношения; Организация (Organization): The MITRE Corporation. Дата изменения (Modification Date): 2017-01-09; Внес изменения (Modifier): CAPEC Content Team, обновлено описание, отношения; Организация (Organization): The MITRE Corporation. Дата изменения (Modification Date): 2017-05-01; Внес изменения (Modifier): CAPEC Content Team, обновлены отношения; Организация (Organization): The MITRE Corporation. Дата изменения (Modification Date): 2018-07-31; Внес изменения (Modifier): CAPEC Content Team, обновлено описание; Организация (Organization): The MITRE Corporation.
	Предыдущее имя записи (Previous_Entry_Name): Дата изменения (Change Date): 2017-01-09; Предыдущее имя записи (Previous_Entry_Name): инъекция (Injection).

В таблице 2.15 представлен шаблон атаки «инъекция трафика» (“Traffic Injection”) [113] из категории «инъекция неожиданных элементов» (“Inject Unexpected Items”). Уровень абстракции данного шаблона — «мета» (в колонке «мета» таблицы 2.13 определено, что подразумевает данный уровень абстракции). Шаблон имеет идентификатор “CAPEC-594”.

Таблица 2.15

Шаблон атаки «инъекция трафика»

Идентификатор шаблона атаки	CAPEC-594
Имя шаблона атаки	Инъекция трафика (Traffic Injection)
Статус (Status)	Стабильная
Описание (Description)	Нарушитель осуществляет инъекцию трафика в сетевое соединение цели. Нарушитель может ухудшить или прервать соединение и потенциально изменить содержимое. Это не флуд, поскольку нарушитель не сосредотачивается на исчерпании ресурсов. Вместо этого он создает определенные входные данные, чтобы повлиять на систему определенным образом.
Идентификатор шаблона атаки	CAPEC-594
Предусловия атаки	Целевое приложение должно использовать открытый канал связи. Канал, по которому общается цель, должен быть уязвим для перехвата (например, атака «человек посередине»).
Требуемые ресурсы	Инструмент, такой как прокси-сервер MITM, который способен генерировать и вводить пользовательские входные данные для использования при атаке.
Результаты атаки (Consequences)	Область действия (Scope): доступность; Ущерб (Impact): ненадежное выполнение.
	Область действия (Scope): целостность; Ущерб (Impact): другое.
Релевантные слабые места	CWE-940 — Некорректная верификация источника канала коммуникации.
Отношения (Relationships) с участниками (Member) и связанными слабыми местами (Related_Weaknesses)	Участник (MemberOf) представления View-1000 (Механизмы атаки, Mechanisms of attack) и представления View-3000 (Области атаки, Domains of attack).
	Предок (ParentOf): стандартного шаблона атаки CAPEC-595.
История содержимого (Content_History)	Представление информации (Submissions): Дата представления (Submission Date): 2017-01-03. Представитель (Submitter): Seamus Tuohy.
	Изменения (Modifications): Дата изменения (Modification Date): 2017-05-01. Внес изменения (Modifier): CAPEC Content Team, обновлены результаты атаки, предусловия атаки, резюме, требуемые ресурсы. Организация (Organization): The MITRE Corporation. Дата изменения (Modification Date): 2019-04-04. Внес изменения (Modifier): CAPEC Content Team, обновлены связанные слабые места. Организация (Organization): The MITRE Corporation.

В таблице 2.16 представлен шаблон атаки «использование состояния гонки» (“Leveraging Race Conditions”) [114] из категории «атаки времени и состояния» (“Time and State Attacks”). Состояние гонки (race condition) — это ошибка проектирования, при которой работа системы или приложения зависит от того, в каком порядке выполняются части кода. Уровень абстракции данного шаблона — «мета» (в колонке «мета» таблицы 2.13 показано, что подразумевает данный уровень абстракции). Шаблон имеет идентификатор “CAPEC-26” и является примером атаки с высоким уровнем критичности. Данный тип атаки подразумевает создание нарушителем состояния гонки для злоупотребления целевым хостом.

Таблица 2.16

Шаблон атаки «использование состояния гонки»

Идентификатор шаблона атаки	**CAPEC-26**
Имя шаблона атаки	Использование состояния гонки (Leveraging Race Conditions)
Статус (Status)	Стабильная
Типичная критичность	Высокая
Описание	Атака направлена на состояние гонки, возникающее, когда несколько процессов получают доступ и оперируют с одним и тем же ресурсом одновременно, и результат исполнения зависит от порядка, в котором происходит доступ.
Идентификатор шаблона атаки	CAPEC-26
Предусловия атаки	Ресурс используется/изменяется одновременно несколькими процессами, так что существует состояние гонки.
	Нарушитель имеет возможность изменить ресурс.
Типичная вероятность применения	Высокая
Уровень необходимых навыков нарушителя	Средний: Способность «запустить гонку» требует базовых знаний параллельной обработки, включая методики синхронизации.
Процесс исполнения (Execution_Flow)	«Разведка» (Explore): Нарушитель исследует, какой уровень доступа у него есть.
	«Эксперимент» (Experiment): Нарушитель получает доступ к ресурсу на целевом хосте. Нарушитель изменяет целевой ресурс. Значение ресурса используется, чтобы определить следующее действие нормального процесса исполнения.
	«Действие» (Exploit): Ресурс изменяется/проверяется одновременно несколькими процессами. Используя один из процессов, нарушитель может изменить значение непосредственно перед тем, как оно будет использовано другим процессом. Возникает состояние гонки, которое нарушитель использует для злоупотребления целевым хостом.

Примеры	Клиент Net Direct client для версий Linux ранее 6.0.5 в Nortel Application Switch 2424, VPN 3050 и 3070, и SSL VPN Module 1000 извлекает и запускает файлы с небезопасными привилегиями, что дает возможность локальным пользователям использовать состояние гонки, чтобы заменить файл в /tmp/NetClient, чтобы другой пользователь вызвал произвольный код при попытке запустить клиент, как показано заменой /tmp/NetClient/client. См. CVE-2007–1057
	. Код ниже иллюстрирует файл, к которому получают доступ по имени множество в открытом каталоге. Состояние гонки существует между доступами, когда нарушитель может заменить файл, на который ссылается имя (см. [REF-107]).
	include <sys/types.h> include <fcntl.h> include <unistd.h> define FILE "/tmp/myfile" define UID 100 void test(char *str) {
	int fd; fd = creat(FILE, 0644); if(fd == -1)
	return;
	chown(FILE, UID, — 1); /* BAD */ close(fd);
	} int main(int argc, char **argv) {
	char *userstr; if(argc > 1) { userstr = argv[1]; test(userstr); } return 0;
	}

Способы минимизации воздействия	Использовать безопасные библиотеки для доступа к ресурсам.

Идентификатор шаблона атаки	CAPEC-26
Способы минимизации воздействия	Осознавать, что неправильное использование таких вызовов функций как chown(), tempfile(), chmod() и т.п. может вызвать состояние гонки.
	Использовать синхронизацию для управления процессом.
	Использовать статический анализ, чтобы найти условия гонки.
	Обращать внимание на проблемы одновременной обработки, связанные с доступом к ресурсам.
Результаты атаки	Область действия (Scope): конфиденциальность, контроль доступа, авторизация; Ущерб (Impact): получение привилегий.
	Область действия (Scope): целостность; Ущерб (Impact): изменение данных.
Идентификатор шаблона атаки	CAPEC-26
Результаты атаки	Область действия (Scope): конфиденциальность, контроль доступа, авторизация; Ущерб (Impact): получение привилегий.
	Область действия (Scope): целостность; Ущерб (Impact): изменение данных.
Релевантные слабые места	CWE-368 — контекстное переключение состояния гонки.
	CWE-363 — разрешение перехода по ссылке на основе состояния гонки.
	CWE-366 — состояние гонки внутри потока и др.
Отношения (Relationships) с участниками (Member) и связанными слабыми местами (Related_ Weaknesses)	Участник (MemberOf) представления View-1000 (Механизмы атаки, Mechanisms of attack) и представления View-3000 (Области атаки, Domains of attack).
	Предок (ParentOf): стандартного шаблона атаки CAPEC-29.
Ссылки	[REF-1] G. Hoglund и G. McGraw. “Exploiting Software: How to Break Code”. Addison-Wesley. February 2004.
	[REF-107] Fortify Software. "SAMATE — Software Assurance Metrics And Tool Evaluation". Test Case ID 1598. National Institute of Standards and Technology (NIST). 2006-06-22. <http://samate.nist.gov/SRD/view_testcase.php?tID=1598 >. И др.
История содержимого (Content_History)	Представление информации (Submissions): Дата представления (Submission Date): 2014-06-23; Представитель (Submitter): CAPEC Content Team; Организация (Organization): The MITRE Corporation.
	Изменения (Modifications): Дата изменения (Modification Date): 2017-01-09; Внес изменения (Modifier): CAPEC Content Team, обновлены связанные шаблоны атаки, тип (*отношение* на *шаблон атаки*); Организация (Organization): The MITRE Corporation. Дата изменения (Modification Date): 2017-05-01; Внес изменения (Modifier): CAPEC Content Team, обновлены зона активации, фазы атаки, предусловия атаки, резюме, вектор инъекции, полезная нагрузка, влияние активации полезной нагрузки; Организация (Organization): The MITRE Corporation. Дата изменения (Modification Date): 2018-07-31; Внес изменения (Modifier): CAPEC Content Team, обновлены требуемые навыки атакующего, примеры, ссылки, способы минимизации воздействия; Организация (Organization): The MITRE Corporation. Дата изменения (Modification Date): 2020-07-30; Внес изменения (Modifier): CAPEC Content Team, обновлены описание, примеры, процесс исполнения, связанные слабые места; Организация (Organization): The MITRE Corporation.

Применение стандарта CAPEC.

Специалисты MITRE определили следующие возможности для использования стандарта CAPEC:

- руководство при определении политик безопасности;
- руководство при создании требований безопасности;
- предоставление контекста для анализа рисков;
- предоставление контекста для тестирования защищенности;
- создание связи между разработкой безопасного программного обеспечения и безопасными операциями.

Одним из примеров использования CAPEC является инструмент, описанный в работе [115], помогающий автоматизировать работу со словарем CAPEC и использовать его при проектировании системы для повышения уровня ее защищенности. В основе работы инструмента лежит формирование отображений между предусловиями системы, шаблонами атак в иерархической форме и стратегиями снижения воздействия атак на исследуемую систему.

Каждому уровню иерархии модели ставится в соответствие элемент схемы CAPEC.

Иерархическая модель включает несколько уровней (с каждым следующим уровнем абстракции увеличивается детализация) (рисунок 2.8).

Рассмотрим эти уровни.

Уровень 1: *уязвимость* (общая классификация для группировки соответствующих ошибок, которые может использовать атакующий). Каждый шаблон атаки CAPEC принадлежит одной из следующих уязвимостей: неправильное использование функциональности (Abuse of Functionality), спуфинг (Spoofing), вероятностные методики (Probabilistic Techniques), использование аутентификации (Exploitation of Authentication), истощение ресурсов (Resource Depletion), использование привилегий/доверия (Exploitation of Privilege/Trust), внедрение (Injection), атаки структуры данных (Data Structure Attacks), атаки утечки данных (Data Leakage Attacks), манипуляция ресурсами (Resource Manipulation), манипуляция протоколами (Protocol Manipulation), атаки времени и состояния (Time State Attacks).

Уровень 2: *шаблон атаки* (высокоуровневый набросок, который описывает различные типы ПО атак).

Уровень 3: *эксплоит* (определенный экземпляр шаблона атаки). Уровень 3.1: *баг/дефект* (используется, чтобы показать различие между проблемой проектирования (недостатком) и проблемой реализации (багом)).

Уровень 4: *зона активации* (область в ПО, способная активировать или запустить эксплоит).

Уровень 5: *вектор внедрения* (реальный формат ввода, используемый при атаке). Уровень 5.1: *спусковой механизм* (ссылается на любые входные данные ПО, чтобы выполнить эксплоит).

Уровень 6: *вознаграждение* (итоговое событие или желаемый результат успешного эксплоита).

На рисунке 2.9 приведен пример иерархической модели для шаблона атаки «SSI-инъекция» (“SSI Injection”), где SSI — Server Side Includes (включения на стороне сервера).

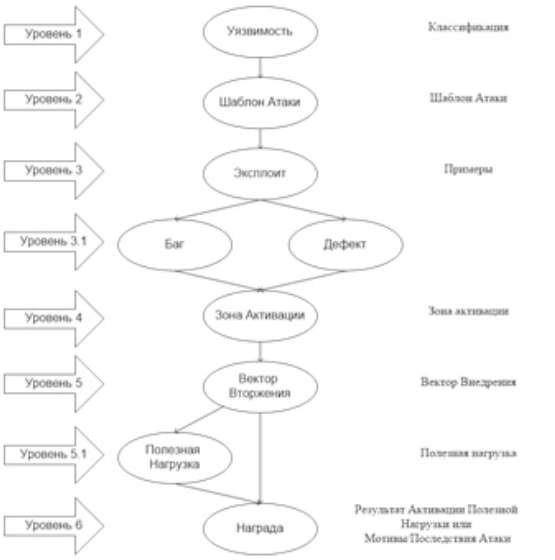


Рис. 2.8. Иерархическая модель и соответствующие элементы CAPEC [115]

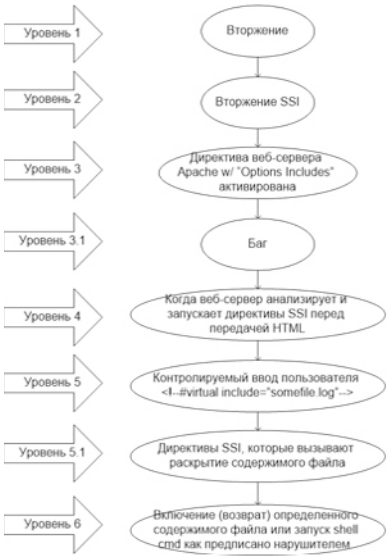


Рис. 2.9. Пример иерархии для шаблона атаки «SSI-инъекция» (“SSI Injection”) [115]

В качестве входных данных инструмент получает предусловия из словаря CAPEC (это решения, принятые при проектировании системы): выбор аппаратуры, операционной системы, конфигурации сервера и языка программирования. После выбора пользователем системных предусловий заполняются соответствующие шаблоны атак и необходимые стратегии минимизации воздействий.

При работе инструмента извлекаются, организуются и редактируются отображения данных, сформированные из предусловий системы, соответствующих шаблонов атак и необходимых стратегий реализации контрмер (рисунок 2.10).

Отображения данных хранятся в базе. Они включают следующую информацию:

1. Системные предусловия:
   - Идентификатор системы.
   - Имя системы.
   - Категория системы.
2. Шаблон атаки:
   - CAPEC-идентификатор шаблона атаки.
   - CAPEC-имя шаблона атаки.
3. Стратегия контрмер:
   - Идентификатор уменьшения воздействия.
   - Имя.
   - Описание.

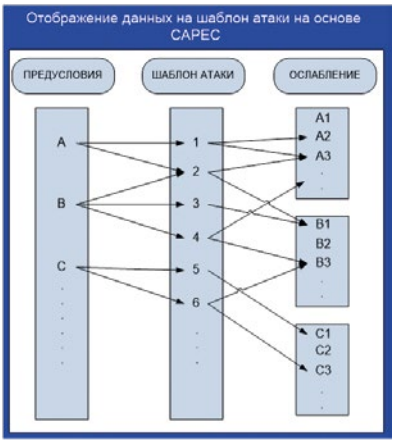


Рис. 2.10. Отображения между предусловиями, шаблонами атак и стратегиями контрмер [115]

Пользователь может редактировать отображения между предусловиями и шаблонами атак, или между шаблонами атак и стратегиями контрмер для конкретной системы.

Результаты, отображающие предусловия системы, соответствующие шаблоны и необходимые стратегии уменьшения воздействий в иерархическом формате представляются графически или в форме таблиц.

Авторы разработали исходный набор отображений данных для многих общих системных предусловий на основе словаря CAPEC в качестве практических указаний. Инструмент позволяет добавлять отображения, редактировать и/или удалять из спецификаций систем.

Системная информация хранится в отдельном наборе таблиц, которые происходят от исходного набора отображений. Это — «профиль системы». Инструмент извлекает и составляет список исходных отображений согласно выбранным предусловиям. Затем эти отображения могут редактироваться пользователем.

В [116] описывается система анализа безопасности OpenSKE (Open Security Knowledge Engineered), основанная на технологии экспертных систем. Авторы используют ряд внешних источников (таких как CVE, CPE, OVAL, CWE, CAPEC) для формирования исходной базы знаний, на основе которой потом при помощи ряда правил прямого логического вывода (вида «ЕСЛИ-ТО») получают новые знания о безопасности системы. В качестве входных данных OpenSKE получает информацию о системе (список всех хостов с учетными записями пользователей, активами, приложениями и т. п.) и найденных уязвимостях (при помощи сканера OVAL). Результатами работы системы являются: список слабых мест (описанных при помощи CWE), список применимых шаблонов атак (CAPEC шаблоны) и список скомпрометированных активов. Шаблоны атак CAPEC моделируются с использованием правил логического вывода на основе предусловий атаки, описанных в шаблоне. Например, для шаблона атаки CAPEC-7 (нарушитель получает доступ к активам в базе данных через скомпрометированное ПО) правило будет выглядеть, как показано на рисунке 2.11. Таким образом, на основе слабых мест системы, предусловий и постусловий шаблонов CAPEC формируются правила логического вывода, позволяющие определить уязвимые активы системы.

В 2015 г. была создана база тактик и методик кибератак MITRE ATT&CK [14]. Выделяются тактики и атаки для предприятий (для платформ Windows, macOS, Linux, PRE, Azure AD, Office 365, Google Workspace, SaaS, IaaS, сеть, контейнеры) и для мобильных технологий (Android, iOS).

Тактики, выделенные для предприятий: разведка (Reconnaisance), включает 10 методик; разработка ресурсов (Resource Development), включает 7 методик; начальный доступ (Initial Access), включает 9 методик; исполнение (Execution), включает 12 методик; закрепление (Persistence), включает 19 методик; увеличение привилегий (Privilege Escalation), включает 13 методик; уклонение от защиты (Defense Evasion), включает 39 методик; доступ к учетным данным (Credential Access), включает 15 методик; открытие (Discovery), включает 27 методик; движение вглубь (lateral movement), включает 9 методик; сбор (Collection), включает 17 методик; командование и контроль

(Command and Control), включает 16 методик; эксфильтрация (Exfiltration), включает 9 методик; ущерб (Impact), включает 13 методик [117].

```
rule "CAPEC-7 : SQL вторжение вслепую"
    when
    # существует нарушитель
    $attacker : User(
            attacker == true
    )
    # ПО, которое может являться целью нарушителя
    $software : Software()
    # ПО содержит любое из перечисленных слабых мест
    exists(
            Weakness(
                    software == $software,
                    identifier in ("CWE-89", "CWE-209",
                                                "CWE-74", "CWE-20",
                                                "CWE-390", "CWE-697",
                                                "CWE-713", "CWE-707")
            )
    )
    # нарушитель может получить доступ к этому ПО
    eval(
            $attacker.getHost().canReach(
                    $software.getHost()
            )
    )
    then
            # нарушитель получает доступ к активам базы данных через данное ПО
            # для имитации было выбрано случайное ПО со случайным типом доступа
            $attacker.addAssetAccess(
                    new AssetAccess(
                            $software.getRandomAsset(AssetType.DATABASE),
                            $attacker,
                            AssetAccessType.getRandomValue()
                    )
            );
            print("[CAPEC-7] Нарушитель '%s' получил '%s' доступ к базу данных
            '%s' через SQL вторжение на ПО '%s'"),
                    $attacker.getFullName(),
                    $attacker.getRecentAssetAccess().getType(),
                    $attacker.getRecentAssetAccess().getAsset().getName(),
                    $software.toString()
            );
    end
```

Рис. 2.11. Спецификация правила логического вывода

Тактики, выделенные для мобильных технологий: начальный доступ (Initial Access), включает 9 методик; исполнение (Execution), включает 4 методики; закрепление (Persistence), включает 9 методик; увеличение привилегий (Privilege Escalation), включает 4 методики; уклонение от защиты (Defense Evasion), включает 19 методик; доступ к учетным данным (Credential Access), включает 11 методик; открытие (Discovery), включает 9 методик; движение вглубь (Lateral Movement), включает 2 методики; сбор (Collection), включает 17 методик; командование и контроль (Command and Control), включает 8 методик; эксфильтрация (Exfiltration), включает 4 методики; ущерб (Impact), включает 10 методик [117]; а также сетевые эффекты (Network Effects), включает 9 методик; и эффекты удаленных сервисов (Remote Service Effects), включает 3 методики [118].

Кроме того, MITRE ATT&CK включает базу способов минимизации воздействия (42 способа) для предприятий и мобильных технологий (13 способов) [119, 120].

Для работы с MITRE ATT&CK разработан язык и формат сериализации Structured Threat Information Expression (STIX™). На языке STIX 2.0 представлен набор данных ATT&CK [121]. Для работы с данным набором данных разработана библиотека на Python [122].

С использованием MITRE ATT&CK построены системы CALDERA™ (система кибербезопасности, разработанная для автоматизации эмуляции нарушителя, помощи “Red Team” и автоматизации реагирования на инциденты) [123], CASCADE (система для автоматизации исследовательской работы, которую выполняла бы “Blue Team” для определения масштабов и вредоносности подозрительного поведения в сети с использованием данных хоста) [124] Cyber Analytics Repository (CAR) (база знаний аналитики) [125].

В отличие от CAPEC, MITRE ATT&CK фокусируется на сетевой защите (в то время как CAPEC фокусируется на безопасности приложений) и описывает фазы эксплуатации жизненного цикла атакующего, действия до и после (например, закрепление, движение вглубь, эксфильтрация), и детализирует отдельные тактики, методики и процедуры, которые используются при целевых атаках (в то время как CAPEC описывает атрибуты и методики, используемые атакующими для применения известных слабых мест) [126].

В настоящий момент протокол SCAP версии 1.3 не предоставляет возможности автоматической реализации защитных мер. Ранее была начата разработка стандартов «Общее перечисление защитных мер» (Common Remediation Enumeration, CRE) и «Расширенная информация по защитным мерам» (Extended Remediation Information, ERI). Хотя разработка была остановлена и стандарты не нашли применения, их можно использовать для создания модели защитных мер, поэтому ниже дается их краткое описание.

2.5. Системы представления защитных мер

Стандарт CRE является схемой определения и описания защитных мер [127]. Каждая защитная мера описывается с помощью CRE entry (элемент CRE). CRE entry представляет собой набор свойств, описывающих защитную меру в формате XML.

Элемент CRE включает:

— CRE-ID (идентификатор CRE) — глобальная уникальная строка, связанная с одним элементом CRE. CRE-ID имеет следующий формат: <PREFIX>:<NAMESPACE>:<ID>-<CHECK-DIGIT>, где PREFIX — текст «cre»; NAMESPACE — имя изготовителя CRE в перевернутом формате системы доменных имен (Domain Name System, DNS), оно не чувствительно к регистру и должно совпадать для всех элементов CRE одного изготовителя; ID — положительное целое число, уникальное в рамках одного NAMESPACE; CHECK_DIGIT — контрольная цифра, вычисленная на основе ID. Пример CRE-ID: cre:gov.exampleagency:5270-4;

— текстовое описание элемента, включающее метод и действие защитной меры;

— список параметров, характеризующих реализацию защитной меры, например, различные виды прав доступа;

— платформу с использованием языка CPE Applicability Language 2.3, для различных платформ CRE элементы должны отличаться;

— ссылки на документацию, описывающую защитную меру;

— метаданные, включающие дату создания элемента, дату изменения элемента, версию и издателя.

Описание элемента CRE не должно меняться и должно быть уникально в рамках CRE-ID NAMESPACE.

Стандарт ERI содержит дополнительную информацию к CRE [128]. ERI включает следующие поля: уникальный идентификатор; ссылка на CRE; индикаторы (ссылки на CCE или CVE); значения, назначаемые параметрам; замены; предусловия; влияние на работу; перезагрузка; метаданные (издатель, действительность).

На текущий момент наиболее известными каталогами защитных мер являются: методический документ ФСТЭК «Меры защиты информации в государственных информационных системах» [54]; каталог средств защиты от NIST, являющийся приложением к документу SP 800–53 Rev. 5 «Средства защиты безопасности и конфиденциальности персональных данных для информационных систем и организаций» [129]; каталог мер защиты MITRE ATT&CK [119, 120]. В документе ФСТЭК [54] меры защиты описываются следующими полями: идентификатор защитной меры; название защитной меры; класс защищенности, к которому применима данная мера; требования к реализации защитной меры; требования к усилению защитной меры; класс защищенности, к которому применима усиленная мера. В каталоге средств защиты от NIST [129] для описания средств защиты используются такие поля: идентификатор средства защиты; название средства защиты; описание средства защиты; обсуждение; идентификаторы связанных средств защиты. В каталоге мер защиты MITRE ATT&CK [119, 120] меры защиты описываются следующими полями: идентификатор защитной меры; название защитной меры; описание защитной меры; версия; дата создания записи о защитной мере; дата последнего изменения записи о защитной мере; техники, против которых направлена защитная мера; ссылки. Стандарты CRE и ERI не были применены на практике для представления защитных мер в рамках описанных каталогов. Однако они стали основой для формирования модели защитной меры, используемой в методике выбора защитных мер, разработанной в СПб ФИЦ РАН и описанной в главе 5.

Выводы по главе 2

К настоящему времени исследователями были разработаны всеобъемлющие протоколы и стандарты представления данных кибербезопасности. Их применение позволит унифицировать представление основных элементов оценивания защищенности и выбора контрмер и автоматизировать данные процессы в рамках систем управления кибербезопасностью.

Глава 3. Методики и средства оценивания защищенности

3.1. Показатели оценивания защищенности

Методики оценивания защищенности позволяют представить уровень защищенности системы от кибератак в форме показателей защищенности.

Согласно [38] показатель — это мера измерения, дающая качественную или количественную оценку определенных атрибутов, выведенную на основе аналитической модели, разработанной для определенных информационных потребностей.

Показатели являются результатом анализа. «Хорошие» показатели должны быть повторяемыми, недорогими при вычислении, численно выраженными, иметь единицы измерения, и соответствовать контексту [130]. Преимущества автоматизации вычисления показателей [130]:

- Точность — точность необходима для доверия собранным данным и для согласованности результатов.
- Повторяемость — также необходимый компонент доверия.
- Надежность — так как в случае автоматизации вычислений нет предрасположенности к ошибкам оператора.
- Прозрачность — шаги получения значений показателей являются явными.

Исследователями были предложены показатели защищенности на основе соответствия критериям, обнаружения вторжений, политик безопасности, инцидентов безопасности и графического моделирования [130]. Это могут быть показатели, вычисляемые на основе информации об инфраструктуре компьютерных систем и сетей, в том числе о составе и топологии компьютерной сети (КС), отражающие характеристики хостов [131] или характеристики приложений [131], показатели, учитывающие информацию об уязвимостях системы и возможных атаках [131]. Ряд показателей рассчитывается на основе графов атакующих действий [130, 132–143] и графов зависимостей сервисов [135, 144–148]. Существуют показатели, которые отражают характеристики защищенности компьютерной системы или сети в целом [138, 149]. Стоимостные показатели определяют стоимость ущерба от атаки и затраты на реагирование [147, 148]. Кроме того, существуют показатели, отражающие возможность атак нулевого дня [150, 151].

Для упорядочивания различных показателей был создан ряд классификаций показателей защищенности. Так, в ряде работ категории показателей выделяются согласно объектам оценки защищенности, например, техническая и организационная категории [152]. В [153] помимо этих двух категорий выделена категория управления. В [154] в классификацию показателей добавлен дополнительный уровень, включающий три категории: безопасность, качество обслуживания и доступность. Для каждой из этих категорий определены технические показатели, организационные показатели и показатели управления. В документе [155] от NIST выделяются показатели реализации (которые используются для демонстрации того,

насколько завершены процессы реализации программ кибербезопасности и связанных с ними политик и процедур), показатели эффективности (которые используются для отслеживания того, насколько правильно реализованы, функционируют и соответствуют требованиям процессы уровня программ и средства управления защищенностью уровня системы) и влияния (которые используются для выражения влияния кибербезопасности на миссию организации).

В [131] помимо разделения на категории (управленческие, операционные, технические), показатели делятся по их функциям для бизнеса: управление инцидентами (насколько хорошо инциденты обнаруживаются, идентифицируются, обрабатываются и исправляются); управление уязвимостями (насколько хорошо определяются известные уязвимости и ослабляется их воздействие); управление заплатками (насколько хорошо поддерживается восстановление системы); управление конфигурациями (состояние конфигурации систем в организации); управление изменениями (как изменения конфигурации системы влияют на ее безопасность); безопасность приложений (позволяет ли модель безопасности работать, как запланировано бизнес-приложениям); финансовые показатели (каков уровень и цель затрат на кибербезопасность).

По способу вычисления показатели делятся на первичные и вторичные [156]. А также выделяются показатели, вычисляемые на основе графов атак (такие, как вероятность атаки, уровень навыков атакующего и другие) и на основе графов зависимостей сервисов (такие, как ущерб от атаки/реагирования, эффективность реагирования и другие) [148].

В [157] выделяется 8 категорий показателей согласно типу значений показателей: существование (показатель того, что что-то существует), порядковое числительное (субъективная качественная оценка), оценка (численные значения для качественных оценок), количественное числительное (число), процент, целое (на основе данных из внешних источников), ценность (на основе потери ценности), неопределенность (включая стохастический и вероятностный аспекты).

Наконец, с точки зрения объекта оценки, можно выделить следующие показатели: показатели, относящиеся к конфигурационным характеристикам системы; показатели, характеризующие атаку; показатели, связанные с защитными мерами; показатели, характеризующие атакующего; показатели, характеризующие уровень защищенности системы в целом.

Для представления уровня защищенности компьютерной системы или сети и выбора защитных мер применяются показатели защищенности, которые являются основой методик оценивания защищенности и выбора контрмер, а также мониторинга защищенности.

Количественные показатели и методики на их основе позволяют измерить защищенность в терминах денежных единиц и частоты нежелательных событий. На основе полученных измерений можно сравнить стоимость рисков и стоимость реализации защитных мер.

Качественные показатели и методики на их основе ранжируют риски относительно друг друга на основе ценности активов, уязвимостей, угроз и защитных мер.

В настоящий момент существует ряд коммерческих и свободно распространяемых продуктов, реализующих методики качественного и количественного оценивания защищенности и выбора защитных мер. Кроме традиционных СМИБ, реализующих методики качественного и количественного оценивания защищенности, для управления кибербезопасностью используются SIEM-системы, предоставляющие множество различных показателей, характеризующих защищенность системы. Существует большое количество научных работ в области формирования и вычисления показателей защищенности и их применения для оценивания защищенности и принятия решений по реализации контрмер и реагированию на инциденты безопасности.

Мониторинг защищенности сети может осуществляться на основе отслеживания и анализа событий, происходящих в системе, путем формирования комплекса показателей и анализа изменений их значений. В частности, мониторинг защищенности осуществляют SIEM-системы. Помимо комплексных показателей, отражающих уровень риска и позволяющих выбрать контрмеры, для мониторинга может применяться ряд вспомогательных показателей, в том числе характеризующих непосредственно элементы конфигурации и безопасности защищаемой системы, такие как система, хосты, программно-аппаратное обеспечение хостов, уязвимости, атакующие действия, угрозы, источники угрозы (атакующие), события безопасности, контрмеры.

Так, в [131] выделяются простые показатели, определяющие количественные, временные и стоимостные характеристики таких элементов безопасности системы, как уязвимости, инциденты и заплатки. Например, *стоимость инцидентов* (*Cost of Incidents*), *процент систем без известных критичных уязвимостей* (*Percent of Systems with No Known Severe Vulnerabilities*), *количество известных уязвимостей* (*Number of Known Vulnerability Instances*), *среднее время устранения уязвимости* (*Mean-Time to Mitigate Vulnerabilities*), *среднее время внедрения заплатки* (*Mean-Time to Patch*) и другие.

Кроме того, в [131] выделяются простые показатели, характеризующие конфигурацию системы и приложения системы с точки зрения защищенности: *количество приложений* (*Number of Applications*), *процент критичных приложений* (*Percentage of Critical Applications), процент покрытия оценки рисков* (*Risk Assessment Coverage*), *ценность приложения* (*Business Application Value*) и другие. *Ценность приложения* определяется как влияние потери хоста на бизнес на шкале «низкая» (потеря конфиденциальности, целостности, доступности будет иметь ограниченный эффект на организацию), «средняя» (потеря конфиденциальности, целостности, доступности будет иметь серьезный неблагоприятный эффект на организацию), «высокая» (потеря конфиденциальности, целостности, доступности будет иметь катастрофический эффект для организации) или «не определена».

В [158] рассматривается показатель, характеризующий способность атакующего выполнить сценарий атаки — *уровень навыков атакующего* (*Attacker Skill Level*, *ASL*). Авторы выделяют пять уровней навыков, в соответствии с уровнем атакующего и его/ее знаниях о системе (таблица 3.1).

Уровень навыков атакующего определяется динамически на основе информации о сложности атакующих действий, полученной через инциденты безопасности:

$ASL = max_i\{ASL_i\}$,

где ASL_i — сложность i-го атакующего действия, $ASL_i \in ExecutedSteps$;

ExecutedSteps — атакующие действия, выполненные данным атакующим.

Таблица 3.1

Пример значений уровня навыков атакующего

ASL	Уровень навыков	Знания о системе
0	Низкий	Нет
1	Средний	Нет
2	Средний	Есть
3	Высокий	Нет
4	Высокий	Есть

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

3.2. Методики и средства качественного оценивания защищенности

Методика оценивания защищенности называется качественной, если в результате ее работы формируется качественная оценка уровня риска. Примерами методик качественного оценивания рисков являются [91]:

- «Облегченный процесс анализа и оценки рисков» (Facilitated Risk Analysis and Assessment Process, FRAAP) [159];
- «Оперативная оценка критических угроз, активов и уязвимостей» (Operationally Critical Threat, Asset, and Vulnerability Evaluation, OCTAVE) [160];
- и др.

FRAAP. FRAAP [159] представляет собой методику качественного оценивания рисков на основе экспертных знаний.

В процессе FRAAP командой внутренних экспертов (бизнес-менеджеров и персоналом) под руководством консультанта определяются возможные угрозы целостности, конфиденциальности и доступности информационных ресурсов. Затем устанавливаются приоритеты угроз, на основе вероятностей их успешной реализации угрозы за заранее определенный период и возможных последствий для безопасности, в форме качественных значений, например, «Высокий»/«Средний»/«Низкий» или «Красный» / «Желтый»/«Зеленый» (пример приведен в таблице 3.2). При этом команда полагается на свои знания об угрозах и уязвимостях.

Таблица 3.2

Определение уровня риска по FRAAP [159]

Вероятность	Ущерб		
	Низкий	Средний	Высокий
Высокая	Желтый	Красный	Красный
Средняя	Зеленый	Желтый	Красный
Низкая	Зеленый	Зеленый	Желтый

После определения угроз и установления относительного уровня риска для каждой угрозы команда определяет средства управления, которые могут снизить риски, концентрируясь на наиболее рентабельных. Эти данные документируются и передаются владельцу, который принимает решение о необходимых средствах управления, учитывая природу ресурса, его критичность для бизнес-операций и стоимость устройства.

Результатом FRAAP является всеобъемлющий набор документов, который определяет угрозы, их приоритеты по уровням риска и возможные средства управления для уменьшения уровней риска угроз.

Достоинства FRAAP:

- использует внутренних экспертов (нет необходимости привлечения внешнего консультанта);
- низкие временные затраты (дни, а не недели или месяцы);
- эффективен по стоимости;
- учитывает бизнес-цели владельцев предприятия (выбранные средства управления направлены на нужды бизнеса).

Недостатки FRAAP:

- команда обычно не пытается получить или разработать показатели защищенности для оценки вероятности угрозы или годовых потерь, за исключением случая, когда данные для определения этих факторов легко доступны.

OCTAVE. Методика OCTAVE [160, 161] предложена институтом Software Engineering Institute при университете Carnegie Mellon. Она представляет собой набор критериев, которые могут использоваться в качестве основы для методики оценки рисков [91]. Критериями определяется процесс, включающий три фазы: построение профилей угроз на основе активов (с использованием деревьев вариантов); идентификация уязвимостей инфраструктуры (с помощью сканеров или вручную); разработка стратегии безопасности и планов (на основе каталогов защитных мер). Методика основана на использовании экспертных знаний и не требует привлечения сторонних экспертов.

3.3. Методики и средства количественного и качественно-количественного оценивания защищенности

В результате применения методики количественного оценивания оценка защищенности формируется в виде количественного значения. К недостаткам точной квантификации рисков относятся следующие аспекты:

• количественные оценки требуют чрезмерных временных затрат для определения и верификации или разработки;
• документация по рискам становится слишком объемной для практики;
• оценки специфических потерь обычно не требуются для определения, нужно ли управление.

Поэтому на практике обычно применяется качественно-количественный подход, когда любому качественному уровню соответствуют определенные диапазоны количественных величин [7]. Данные методики рассмотрены в этом разделе, для качественно-количественных значений указываются качественные уровни и соответствующие диапазоны количественных величин.

Методика на основе карты сети. В [162] оценка риска определяется на основе карты достижимости хостов. Предлагаются следующие показатели для хостов: *ценность для бизнеса* (*Business Value*), определяемый на шкале от 0 до 100 как ценность самого важного сервиса хоста; *уязвимость хоста* (*Exposure Score*) — это оценка возможности того, что хост будет атакован, *оценка риска* (*Risk Score*), *унаследованный риск* (*Downstream Risk Score*) — суммарный риск для тех хостов, которые можно атаковать с данного хоста.

Уязвимость хоста определяется на шкале от 0 до 1 на основе достижимости хоста и простоты эксплуатации уязвимостей хоста. Входными данными для вычисления данного показателя являются временная оценка CVSS для каждой уязвимости хоста и контекст хоста на карте угроз (т.е. достижимость хоста). Уязвимость хоста X определяется по формуле: $Exposure(X) = \max_i Exposure(A_i) \cdot Exposure(A_i, X)$, где A_i — i-й предок хоста X; $Exposure(A_i) \cdot$– уязвимость предка, определяемая на основе максимальной временной оценки CVSS уязвимостей хоста предка; $Exposure(A_i, X)$ — временные оценки уязвимостей хоста X, доступных с хоста A_i.

Оценка риска определяется на основе показателей *ценность для бизнеса* и *оценка подверженности воздействию* на шкале от 0 до 100.

Унаследованный риск вычисляется обходом карты достижимости хостов снизу вверх, в обратном порядке относительно вычисления *уязвимости хоста.*

Методика на основе иерархии объектов сети. В [134] выделяется несколько уровней оценивания: уровень уязвимостей, уровень сервисов, аппаратный уровень и сетевой уровень. Уровень риска сети определяется на сетевом уровне.

Вначале определяются уровни рисков сервисов и хостов на соответствующих уровнях оценивания.

Для определения коэффициента риска i-го сервиса вначале определяется его критичность для сети SI_i на основе частоты обращения к сервису (авторы выделяют три уровня критичности: 3 — высокая, 2 — средняя и 1 — низкая, которые назначаются вручную сетевым администратором). Затем определяется влияние уязвимостей на сервис s_i: $ln \sum_{j=1}^{m} e^{100 \times vs_{ij} \times p_j}$, где m — количество уязвимостей, влияющих на доступность сервиса s_i; vs_{ij} — влияние уязвимости v_j на сервис s_i; p_j — вероятность успешной эксплуатации уязвимости v_j. Коэффициент риска для сервиса s_i определяется по формуле: $SR_i = SI_i \times ln \sum_{j}^{m}{}_{1} e^{100 \times vs_{ij} \times p_j}$.

На аппаратном уровне определяется коэффициент риска хостов. Каждому хосту присваивается уровень критичности *HI* (авторы выделяют следующие уровни критичности: 1, 0,8, 0,6, 0,4, 0,2, где 1 — максимальный уровень важности).

Коэффициент риска для конфиденциальности хоста h_i определяется по формуле: $CR_i = ln \sum_{j=1}^{m} e^{100 \times vc_{ij} \times p_j}$, где m — количество уязвимостей, влияющих на конфиденциальность хоста; vc_{ij} — влияние уязвимости v_j на конфиденциальность h_i; p_j — вероятность успешной эксплуатации уязвимости v_j. Аналогично можно получить коэффициент риска для целостности хоста.

Коэффициент риска для k-го хоста определяется по формуле:

$$HR_k = \left(a_s \sum_{j=1}^{n} SR_j + a_c CR_k + a_i IR_k\right) \times HI_k,$$

где n — количество сервисов хоста;

a_s, a_c, a_i — веса доступности, конфиденциальности и целостности, устанавливаемые на основе актуальной сетевой операции сетевыми администраторами, $a_s + a_c + a_i = 1$.

После нормализации можно получить отношение риска хоста h_k к риску всей сети:

$HT_i = HR_i / \sum_{k=1}^{n} HR_k$.

На сетевом уровне значения риска всех устройств сравниваются с максимальным и минимальным риском в сети.

Максимальное значение риска в сети определяется по формуле:

$NR_{max} = \sum_{k=1}^{n} R_{i_k} HI_{j_k}$,

где $R = \{R_{i_1}, R_{i_2}, ..., R_{k_n}\}$ — возрастающий набор значений риска без учета степени важности хостов,

$HI = \{HI_{i_1}, HI_{k_2}, ..., HI_{i_n}\}$ — возрастающий набор степеней важности хостов.

Минимальное значение риска в сети определяется по формуле: $NR_{min} = \sum_{k=1}^{n} R_{i_k} HI_{j_{n-k+1}}$.

Реальное значение риска определяется по формуле: $NR = \sum_{i=1}^{n} HR_i$.

Выводы о защищенности сети делаются на основе степени распределения сетевого риска $NRD = (NR - NR_{min})/(NR_{max} - NR_{min})$, $0 \le NRD \le 1$, $0 \le NRD \le 1$.

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

Можно выделить следующие типы графов атак:

• Полный граф атак [163] — включает все пути, которыми атакующий может скомпрометировать сеть. Недостаток: сложность $O(n!)$, сложность слишком быстро растет с ростом размера сети.

• Предиктивный граф [164] — узел добавляется в граф в том случае, если ни один предок данного узла не использует ту же уязвимость для перехода в то же состояние, что и новый узел. Данные графы строятся намного быстрее, но все еще содержат лишние структуры и не позволяют учитывать учетные записи.

• Граф со множеством предусловий [165] — включает три типа узлов: состояние (уровень доступа атакующего на хосте); предусловия (достижимость или учетные записи); уязвимость (отдельная уязвимость). Дополнительные циклические дуги добавляются для отображения связей с уже существующими узлами. Преимуществом данного графа является оперативность построения. Его можно преобразовать в полный или предиктивный граф.

При работе с графами атак можно выделить следующие проблемы:

— высокое время обработки, связанное с тем, что размер графа экспоненциально растет с увеличением количества уязвимостей;

— проблема обработки циклов.

Ряд исследований посвящен решению первой проблемы. При построении графов атак предполагается, что они обладают свойством монотонности, что позволяет уменьшить их сложность с экспоненциальной до полиномиальной [130, 165]. Кроме того, решение проблемы оперативного построения графов рассматривается в работе [166].

Второй вопрос рассматривается в [132, 139, 140].

В [132] и в [140] предполагается, что для атакующего не имеет смысла повторное посещение уже посещенных узлов.

В [139] выделяется три вида циклов для графов атак: циклы, которые могут быть полностью удалены (в случае, если никакие узлы цикла не могут быть достигнуты); циклы, которые могут быть безопасно разорваны (в случае, если узлы цикла могут быть достигнуты); циклы, которые не могут быть удалены или разорваны.

Когда циклы существуют в графе атак, нарушитель может достигнуть действия или условия более чем один раз. Чтобы избежать повторного учета доли атакующих действий, которые уже были учтены, при расчете вероятности успешности действия или условия, удаляются все исходящие дуги узла, при этом граф обновляется таким образом, чтобы удалить недостижимые вершины и дуги.

Для учета циклов авторы [139] предлагают следующее определение графа атак: пусть дан граф атак $G = (E \cup C, R_r \cup R_i)$, где E — множество узлов, C — множество дуг, $R_r \subseteq C \times E$, $R_i \subseteq E \times C$, и любая функция присвоения условных вероятностей p: $E \cup C \rightarrow [0,1]$. Пусть $A(G,e)$ (или $A(G,c)$) — граф атак, полученный удалением из G всех исходящих дуг от e (или c) и, соответственно, удалением всех недостижимых действий и условий из G. Тогда функция присвоения полных вероятностей P: $E \cup C \rightarrow [0,1]$ определяется следующим образом:

• если e (или c) не появляется в $A(G,e)$ (или $A(G,c)$), то $P(e)=0$ (или $P(c)=0$);

• иначе $P(e)$ (или $P(c)$) равно его значению, вычисленному на $A(G,e)$ (или $A(G,c)$).

При определении вероятностей всех узлов (вероятности успешности действий или условий) применяется модифицированный обход в ширину: значение для вершины вычисляется только после того, как посчитаны значения для всех ее предков. В случае если достигнут цикл (то есть не могут быть посчитаны значения для всех предков узла), основная процедура останавливается, и запускается процедура, вычисляющая вероятность вершины в соответствии с определением выше. Алгоритм представлен на рисунке 3.1. В основном цикле (строки 3-9) вычисляются оценки полной вероятности для всех вершин, которым изначально не присвоено значение вероятности 1. Каждый цикл включает: (1) вычисление вероятностей, пока не встретился цикл на основе модифицированного обхода в ширину (строки 4-5); (2) вычисление вероятностей в циклах для вершин, у которых более одной входящей дуги на основе определения выше (строки 6-7); (3) вычисление общих оценок в цикле для вершин с одной входящей дугой так, как если бы они не были в цикле (строки 8-9).

На основе графов атак был разработан ряд вероятностных моделей. Для учета неопределенности того, какое именно атакующее действие будет выполнено и насколько успешно, в ряде работ предлагается использовать вероятностные графы атак для анализа защищенности системы [130, 134–136], в других работах применяются байесовские графы атак [133, 137–143].

Ввод: граф атак G с условными всроятностями для всех всршин
Вывод: набор полных всроятностей для всех всршин G
Методика:
1. **Для** каждого изначально удовлстворснного условия c
2. **Присвоить** $P(c) = 1$ и обозначить c как обработаннос
3. **Пока** сущсствуют нсобработанные всршины
4. **Пока** сущсствуст нсобработанная всршина v чьи прсдки обработаны
5. **Вычислить** $P(v)$ и обозначить v как обработанную
6. **Для** каждой всршины v' цикла, имсющсй большс одной входящсй дуги
7. **Вычислить** $P(v')$ и обозначить v' как обработанную
8. **Для** каждой нсобработанной всршины v'' в циклах
9. **Вычислить** $P(v'')$ и обозначить v'' как обработанную
10. Всрнуть набор полных всроятностей для всех всршин

Рис. 3.1. Алгоритм вычисления вероятности успешности действий или условий на графе атак [139]

Методики на основе вероятностных графов атак. Вероятностный граф атак отражает все возможные пути компрометации системы и показывает распространение вероятности атаки [130].

В [150] предлагается методика расчета ущерба от атаки APr:

$$APr(D) = \sum_{d \in D} P(d) SBE_d,$$

где D — все хосты сети, которые могут быть начальной точкой атаки;

$P(d)$ — вероятность того, что на хосте d есть уязвимость;

SBE_d — ущерб, который будет нанесен в случае компрометации хоста d.

SBE определяется на основе графа, в котором узлами являются хосты, а дуги — сервисами, которые связывают хосты. *SBE* определяется по формуле:

$$SBE_d = \sum_{n \in N} \left(\prod_{m \in M} p_{s_{dm}} \right) Cost_n,$$

где N — количество хостов в сети;

M — количество хостов, связанных с d;

$p_{s_{dm}}$ — мера уязвимости сервиса s, вычисляемая на основе критичности уязвимости для сети и насколько уязвим был сервис в прошлом;

$Cost_n$ — ущерб.

В [167] предлагается подход к анализу защищенности, состоящий из следующих этапов:

- построение графа атак;
- анализ графа атак и оценивание защищенности.

Предлагаемая в [167] качественно-количественная методика оценивания общего уровня защищенности компьютерной сети базируется на индексах CVSS и применении некоторых процедур методики анализа рисков FRAAP.

Методика оценивания общего уровня защищенности включает следующие этапы:

- вычисление показателей защищенности базовых и составных объектов общего графа атак (*Criticality* — критичность хоста, *Severity* — уровень критичности атакующего действия, *AccessComplexity* — сложность доступа к уязвимости, *Realization* — степень возможности реализации угрозы);
- получение качественных оценок уровня риска для всех угроз (*RiskLevel*);
- оценивание уровня защищенности анализируемой компьютерной сети (*SecurityLevel*) на основе полученных оценок уровней риска всех угроз.

В [132] предлагается методика количественного оценивания защищенности на основе графа привилегий, преобразованного в Марковскую цепь. Граф привилегий представляет собой набор узлов (каждый узел отображает набор привилегий пользователя) и дуг, отображающих возможность расширения привилегий за счет эксплуатации уязвимостей. Для отображения переходов между состояниями в процессе проведения атаки, граф привилегий преобразуется в сеть Петри. Переход из одного состояния в другое происходит при получении новых привилегий.

Атаки оцениваются с точки зрения двух параметров: времени, необходимого на успешную реализацию атаки (защищенность прямо пропорциональна времени, необходимому на успешную реализацию атаки), и затрат, необходимых для ее реализации (безопасность прямо пропорциональна усилиям, требуемым для проведения атаки). Для оценки времени и затрат, требуемых на атаку, применяется Марковская модель. В соответствии с ней, вероятность успеха атаки до времени t описывается экспоненциальным распределением: $P(t)=1 - \exp(-\lambda \cdot t)$, где λ — параметр распределения. $P(t)$ монотонно возрастающая функция, которая изначально равна 0 и асимптотически достигает 1.

На основе применения Марковской модели выводятся и другие показатели, включая *среднее время до повреждения MTTF*. Данный показатель характеризует среднее время достижения цели атакующим (чем выше *MTTF*,

тем лучше безопасность). Для графа состояний *MTTF* при начальном состоянии k задается суммой среднего времени остановок в состояниях, ведущих к цели, взвешенного вероятностью перехода в данные состояния:

$$MTTF_k = T_k + \sum_{l \in out(k)} P_{kl} \times MTTF_{ld},$$

где $P_{ld} = \lambda_{kl} \times T_k$ — условная вероятность перехода из состояния k в состояние l;

T_k — среднее время остановки в состоянии k;

$out(k)$ — состояния, в которые можно перейти из состояния k;

λ_{kl} — коэффициент перехода из состояния k в состояние l.

В [136] предлагается простой показатель уверенности в атаке *Confidence_level*, который определяет уровень уверенности в том, что атака находится в процессе выполнения. Он рассчитывается по следующей формуле:

$$Confidence\_level = \frac{number\_of\_sequence\_occurences}{total\_number\_of\_sequences\_with\_this\_prefix},$$

где $number\_of\_sequence\_occurences$ — количество повторений шаблона атаки;

$total\_number\_of\_sequences\_with\_this\_prefix$ — общее количество атак с соответствующим префиксом на графе атак.

В [135] для определения уровня уверенности в том, что узел графа атак был достигнут, используется индекс уверенности в компрометации (*Compromised Confidence Index*, *CCI*). Индекс устанавливается в единицу, если детектор, сгенерировавший предупреждение об атаке, не передает значения доверия. Для родительских узлов (графа атак) он определяется на основе значения доверия предупреждения, соответствующего узлу графа и показателей *CCI* для прямых потомков:

$$CCI = \begin{cases} alert\_confidence, & \text{нет потомков} \\ f'(CCI_i), & \text{нет детектора} \\ f\left(f'\left(CCI_i, alert_{confidence}\right)\right), & \text{иначе} \end{cases},$$

где i — номер потомка;

CCI_i — значение *CCI* для i-го потомка;

$alert\_confidence$ — значение доверия детектора;

$$f' = \begin{cases} \max(CCI_i), & \text{связь типа ИЛИ} \\ \min(CCI_i), & \text{связь типа И} \\ mean(CCI_i | CCI_i > \tau), & \text{Кворум достигнут} \\ 0, & \text{Кворум не достигнут} \end{cases},$$

τ — порог, задаваемый для каждого узла.

Значение $alert\_confidence$ устанавливается в 1, если оно не передается детектором, в противном случае оно вычисляется по формуле:

$$alert\ confidence = alert\ confidence \times (1 - false\ alarm\ probability),$$

где $false\_alarm\_probability$ — вероятность того, что предупреждение ложное.

Изначально $false\_alarm\_probability$ равна 0. При получении предупреждений от детекторов показатель пересчитывается по формуле:

$$false\_alarm\_probability = \alpha \times links\_probability + (1 - \alpha) \times history\_probability,$$

где $links\_probability$ — параметр, который определяет отсутствие очевидной связи предупреждения с другими предупреждениями;

$history\_probability$ — историческая вероятность;

α — параметр смещения вероятностной кривой, который зависит от текущей и предыдущих вероятностей связи.

Параметр $links\_probability$ определяется из уравнения:

$$1 - \max(links\_probability) = 1 - \max(temporal\_link\_probability, spatial\_link\_probability)$$

где $spatial\_link\_probability$ — вероятность пространственной связи;

$temporal\_link\_probability$ — вероятность временной связи;

Вероятность пространственной связи определяется по формуле:

$$spatial\_link\_probability = 1/(1 + \gamma_q \times q),$$

где γ_q — масштабирующий параметр;

q — минимальное пространственное расстояние между предупреждениями в одном подграфе атаки.

Вероятность временной связи определяется по формуле:

$$temporal\_link\_probability = 1/(1 + \gamma_p \times p),$$

где γ_p — масштабирующий параметр;

p — минимальное временное расстояние между текущим предупреждением и предупреждениями, возникавшими на предыдущих итерациях, которые происходили пространственно близко к текущему предупреждению.

Историческая вероятность рассчитывается с учетом предыдущих вероятностей связи по формуле:

$$history\_probability = \beta \times links\_probability + (1 - \beta) \times history\_probability,$$

где β — параметр смещения вероятностной кривой, который зависит от текущей и предыдущих вероятностей связи.

В [134] предлагается иерархический метод количественной оценки риска на основе графа атак. Авторы выделяют следующие уровни сети: уровень уязвимостей, уровень сервисов, аппаратный уровень и сетевой уровень.

На уровне уязвимостей вычисляются вероятности успешного использования уязвимостей, соответствующих узлам графа атак.

Уязвимость *v* определяется следующим образом набором параметров:

$$v=\{vid, Sp, Dp, Rp, Conn, Co, Ca, I, C, A\},$$

где *vid*, *Sp*, *Dp*, *Rp*, *Conn* — параметры, необходимые для генерации графа атак, *vid* — идентификатор уязвимости, *Sp* — уровень привилегий на исходном хосте, необходимый для использования уязвимости, *Dp* — уровень привилегий на целевом хосте, *Rp* — уровень привилегий, получаемый на целевом хосте после атаки, *Conn* — протокол связи между исходным и целевым хостом, необходимый для использования уязвимости;

Co, *Ca*, *I*, *C*, *A* — параметры, необходимые для оценки защищенности, *Co* — сложность атаки (авторы выделяют пять уровней сложности: 1, 0.8, 0.6, 0.4, 0.2, где оценка 1 назначается самой простой для реализации атаке), *Ca* — категория уязвимости (0 — атака на отказ в обслуживании, 1 — атака на данные, 2 — атака на повышение привилегий), *I*, *C*, *A* — влияние на целостность, конфиденциальность и доступность, соответственно.

Вероятность того, что на j-м шаге атакующий выберет уязвимость v_j, вычисляется по формуле:

$$\lambda_{ij} = Co_{ij} / \sum_{k=1}^{n} Co_k,$$

где C_{0ij} — сложность атаки с использованием уязвимости v_j,

$\sum_{k=1}^{n} Co_k$ — сумма сложностей атак с использованием всех остальных доступных уязвимостей.

Вероятность успешной атаки с узла n_i на целевой узел графа атак за m шагов рассчитывается итеративно по следующей формуле:

$$p_i^m = \sum_{n_j \in Q(n_i)} \lambda_{ij} \times Co_{ij} \times p_j^{m-1},$$

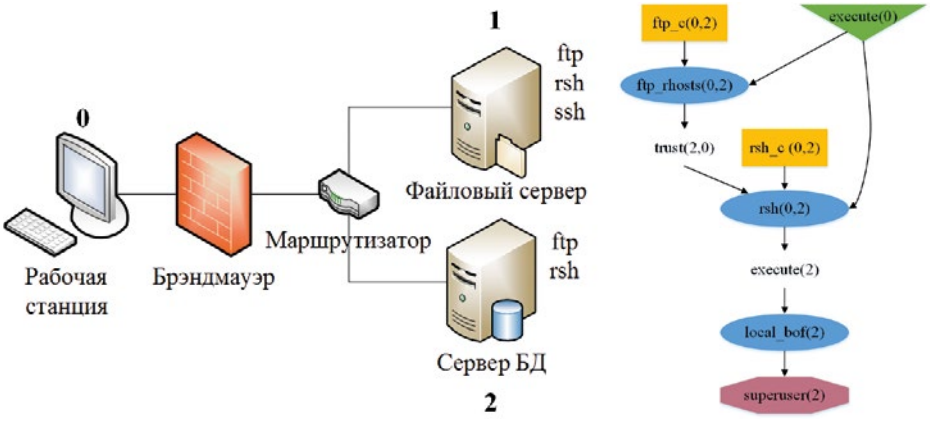
где n_j — атакуемый узел,

$Q(n_i)$ — множество всех узлов, доступных с n_i.

Процесс заканчивается, когда $p_j^{m-1} = 1$.

На последующих уровнях вычисляется уровень риска для сервисов, хостов и сети.

В [130] рассматривается простая методика анализа рисков на основе вероятностного графа атак, позволяющая определить наиболее вероятные пути атаки в сети. Авторы используют граф, предложенный в [168], который можно отнести к типу графов со множеством предусловий (рисунок 3.2). Фрагмент графа, отражающий атаку на сеть, представленную на рисунке 3.2(а), приведен на рисунке 3.2(б). На рисунке 3.2(а) цифры, выделенные жирным шрифтом (0, 1 и 2) обозначают номера хостов. Граф отражает связи между конфигурацией системы, то есть предусловиями атакующих действий (желтые прямоугольники на рисунке 3.2(б), зеленый треугольник отображает исходные возможности атакующего), атакующими действиями (голубые овалы на рисунке 3.2(б)) и последствиями атакующих действий (простой текст на рисунке 3.2(б), конечная цель атакующего отображена в красном восьмиугольнике). На рисунке 3.2(б) цифры в круглых скобках обозначают хост-источник атаки и целевой хост.



а) б)

Рис. 3.2. Пример вероятностного графа атак [130]

В работе предлагается два типа показателей — индивидуальные показатели (характеризуют свойства отдельных компонентов графа, не учитывают отношения между компонентами) и интегральные показатели (учитывают отношения между компонентами). Для вычисления индивидуальных показателей применяются индексы CVSS, полученные из открытой базы уязвимостей NVD [13]. В качестве индивидуального показателя рассматривается условная вероятность успешной атаки в случае, если соблюдены все предусловия (для этого используется индикатор CVSS «вектор доступа»). В качестве интегрального показателя рассматривается абсолютная вероятность успешной атаки, вычисляемая на основе перемножения индивидуальных показателей (так как чем больше шагов необходимо выполнить для успешной реализации атаки, тем меньше вероятность ее успеха). Для выбора защитных мер интегральные показатели пересчитываются с учетом введения отдельных защитных мер. Выбираются те меры, которые позволяют снизить интегральный показатель вероятности успешной атаки на наиболее ценные хосты.

Тем не менее в работе не раскрыты методики решения проблем с циклами на графе; при расчете интегральных показателей не учитывается ущерб, наносимый атакующим действием; не предложено автоматизированной методики выбора защитных мер.

Методики на основе Байесовских графов атак. Байесовский граф — это граф, узлам которого соответствуют случайные переменные Бернулли, отображающие состояние узла (скомпрометирован или нет).

В [139] авторы предлагают показатель вероятности многошаговых атак, использующих комбинации различных уязвимостей, вычисляемый на основе графа атак. В данной работе граф атак представляет собой множество вершин (отображающих действия или условия, необходимые для перехода к следующему действию) и дуг.

Каждому узлу, отображающему действие e, соответствует численное значение $p(e)$ — относительная вероятность того, что соответствующее действие будет выполнено нарушителями, когда все требуемые условия соблюдены. Это значение зависит от соответствующих уязвимостей и присваивается на основе знаний экспертов об используемой уязвимости. На практике индивидуальные оценки могут быть получены преобразованием оценок уязвимостей, предоставленных существующими стандартами, вроде CVSS, в вероятности.

Каждому узлу, отображающему условие c, соответствует численное значение $p(c)$, равное 1.

На основе данных вероятностей рассчитываются значения $P(e)$ и $P(c)$ — полные вероятности того, что атакующий может успешно достичь узла и выполнить действие или удовлетворить условие на заданном графе атак, соответственно. При этом предполагается независимость событий осуществления действий атакующим друг от друга. Кроме того, учитываются отношения между действиями и условиями: конъюнкция (существует между несколькими условиями, требуемыми для выполнения одного действия) и дизъюнкция (существует между несколькими действиями, которые удовлетворяют одному условию).

В соответствии с этим вероятность успешного выполнения атакующего действия вычисляется по формуле:

$$P(e) = p(e) \cdot \prod_{c \in R_r(e)} P(c),$$

где $R_r(e)$ — множество условий, которые необходимо удовлетворить для успешного осуществления действия e.

Вероятность успешного удовлетворения условия определяется по формуле:

$$P(c) = p(c), \text{ если } R_i(c) = \varnothing;$$

$$P(c) = p(c) \cdot \bigotimes_{e \in R_i(c)} P(e), \text{ иначе,}$$

где $R_i(c)$ — множество действий, которые удовлетворяют условию c;

$\bigotimes$ — оператор, который рекурсивно определяется как $\bigotimes P(e) = P(e)$ для любого $e \in E$, E — множество узлов графа, соответствующих действиям, и $\bigotimes (S_1 \cup S_2) = \bigotimes S_1 + \bigotimes S_2 - \bigotimes S_1 \cdot S_2$ для любых дизъюнктных и непустых наборов $S_1 \subseteq E$ и $S_2 \subseteq E$.

Для вычисления вероятностей всех узлов графа применяется обход в ширину.

Вероятность $p(e)$ можно интерпретировать как долю нарушителей, которые могут и исполнят e, среди всех нарушителей, которые попытаются скомпрометировать заданную сеть за заданное время. При этом учитываются два фактора: есть ли у нарушителя знания и ресурсы для выполнения e, и выберет ли он/она e для выполнения. Тогда $P(e)$ и $P(c)$ обозначают долю нарушителей, которые будут успешно использовать e или удовлетворять c, соответственно, в заданной сети.

В [138] графы атак применяются для вычисления уровня уязвимости критических ресурсов. Предлагаемая методика включает: (1) создание профиля атакующего (включая уровень навыков, позицию и время); (2) формирование графов атак (узлы соответствуют действиям, дуги — связям между ними); (3) назначение поведенческих атрибутов узлам графа атак (уровень навыков, позиция и время) и формирование профильных графов атак в соответствии с профилем атакующего; (4) вычисление вероятности компрометации узлов графа на основе Байесовского вывода: с учетом исходных вероятностей компрометации (статистические данные) и вероятности компрометации узла, если скомпрометированы его предки, вычисляются апостериорные вероятности компрометации.

В [169] рассматривается более сложный вероятностный подход с использованием исторических данных, реализованный на основе Байесовского вывода. Авторы рассматривают вычисление таких показателей, как *уровень навыков атакующего* и *вероятность атаки*.

Вначале каждой уязвимости $v \in ,V$ где V — множество всех уязвимостей системы, назначается *требуемый уровень навыков* k_v, который может принимать значения из набора $z(v) = \{z_0, z_1, z_2, z_3\}$, где z_0 — низкий уровень навыков, z_1 — средний низкий уровень, z_2 — средний высокий уровень, z_3 — высокий уровень. Затем реальный уровень атакующего симулируется как нормальное распределение над требуемым уровнем навыков на отрезке значений [0, 1]:

$$p(k_v|m_z,\sigma_z) = \frac{1}{\sigma_z\sqrt{2\pi}} e^{-\frac{(k-m_z)^2}{2\sigma_z^2}},$$

где m_z, σ_z — параметры распределения (математическое ожидание и дисперсия).

В [169] параметры распределения определены следующем образом: для $\forall i \in \{0,1,2,3\}$: $m_{z_i} = \frac{i}{4} + \frac{1}{8}, \sigma_{z_i} = \frac{1}{5}$.

Вероятность успешной эксплуатации уязвимости определяется на основе функции распределения:

$$\Phi(k,z) \propto \int_k^1 p(k_v|m_z,\sigma_z)dk_v = \text{erf}\left(\frac{k-m_z}{\sigma_z\sqrt{2}}\right) - \text{erf}\left(\frac{-m_z}{\sigma_z\sqrt{2}}\right),$$

где $\text{erf}(k) = \frac{1}{\sqrt{2\pi}} e^{\frac{k^2}{2}}$ — функция ошибок.

Вероятность успешной атаки определяется на основе следующей методики. Пусть $P_e(d|a_0,s,X,\mathcal{A})$ — вероятность хотя бы одной успешной атаки на данные d, с учетом исторических данных об атаках X за следующие s временных интервалов, где $a_0 \in \mathcal{A}$ — начальное состояние графа атак $\mathcal{A}$, $x \in X$ — экземпляр атаки, $e(x) \in \{0\ 1\}$ — приравнивается 1, если атака была успешной, 0 — если нет.

Ожидаемое количество атак определяется на основе апостериорного распределения, с учетом уровня навыков атакующего k и параметра Пуассоновского распределения λ:

$$P(k,\lambda|X) \propto P(X|k,\lambda) \cdot P(k) \cdot P(\lambda),$$

где $P(k)$ — априорное распределение для k, авторы [169] используют равномерное распределение, $P(k) \propto 1$.

$P(\lambda)$ — априорное распределение для λ, авторы [169] используют Гамма-распределение.

$P(X|k,\ \lambda)$ — функция, определяющая вероятность того, что множество атак X было проведено группой атакующих с уровнем навыков k и параметром λ. Данная вероятность определяется по формуле:

$$P(X|k,\lambda) = P(X|k) \cdot P(n(X)|\lambda),$$

где $P(X|k)$ — вероятность X при заданном k, $P(X|k) \propto \prod_{x\in X} P(x|k)$, где $P(x|k)$ — вероятностное распределение над x при заданном $k \in [0,1]$: $P(x|k) = (1-\Phi(k,z_i))^{e_{-z_i}} \cdot (\Phi(k,z_i))^{e_{z_i}}$, где e_{-z_i} — общее число безуспешных атак на уязвимости с требуемым уровнем навыков z_i, e_{z_i} — общее число успешных атак на уязвимости с требуемым уровнем навыков z_i.

$P(n(X)|\lambda)$ — вероятность того, что за время T произойдет $n(X)$ атак, которая определяется на основе распределения Пуассона с математическим ожиданием $n(T) \cdot \lambda$:

$$P(n(X)|\lambda) = \frac{(n(T)\lambda)^{n(X)} e^{-n(T)\lambda}}{n(X)!}.$$

Вероятность хотя бы одной успешной атаки определяется маргинализацией (накоплением всех условных вероятностей). Пусть $P_e(d|a_0,\mathcal{A},k)$ — вероятность, что атакующий с уровнем навыков $k \in [0,1]$

успешно проэксплуатирует уязвимости графа $\mathcal{A}$, и набор данных d будет скомпрометирован. Данная вероятность вычисляется методом Монте-Карло. Тогда вероятность хотя бы одной успешной атаки из n определяется как: $P_e(d^n|a_0,\mathcal{A},k) = 1-(1-P_e(d|a_0,\mathcal{A},k))^n$. Вероятность хотя бы одной успешной атаки на набор данных d с учетом X за следующие s временных интервалов вычисляется по формуле:

$$P_e(d|a_0,s,X,\mathcal{A}) \propto 0\int_0^1\int_0^\infty P_e(d^n|a_0,\mathcal{A},k)P(n(s)|\lambda)P(X|k,\lambda)p(k)p(\lambda)d\lambda dk = C -$$

$$-\int_0^1 \frac{1}{\left(1+\frac{sP_e(d|a_0,\mathcal{A},k)}{b+n(T)}\right)^{a+n(X)}}\prod_{i=0}^{3}(1-\Phi(k,z_i))^{e-z_i}\cdot(d^n|a_0,\mathcal{A},k)\cdot\Phi(k,z_i)^{e_{z_i}}dk,$$

где $n(s)$ — количество атак за следующие s временных интервалов, для которого применяется распределение Пуассона с математическим ожиданием $\lambda{\cdot}s$,

C — константа,

a, b — параметры Гамма-распределения.

В [149] рассматривается автоматическая генерация байесовского графа атак и оценивания защищенности сети на основе анализа деревьев недочетов. Авторы предлагают три показателя, основанных на анализе деревьев недочетов, которые позволяют определить уровень защищенности сети (и потом применять его для определения изменения уровня защищенности после внедрения защитных мер): *ненадежность конечного события* (*Unreliability of the Top Event*, *TE*); *критичность начальных событий* (*Criticality of Bottom Events*, *BEs*), то есть начальных событий байесовского графа атак; *наиболее критичный компонент системы* (*The Most Critical System Component*), то есть наиболее критичная уязвимость.

Ненадежность конечного события определяется следующим образом:

$$P(TE=1) = p(s_s=1) = \sum_{s_1,\dots s_k} P(s_1,\dots,s_k,s_s=1),$$

где s_s — успешное состояние (для атакующего);

$s_1,\dots,s_k$ — набор начальных состояний, $1 \le k \le s$.

Критичность начальных событий определяется следующим образом:

$$P(s_0^k|TE=1),\ 1 \le k \le j,$$

где s_k^0 — набор начальных состояний.

Наиболее критичный компонент системы определяется на основе значения критичности:

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

где v_k — уязвимость системы, $v_k \in V$;

$V=\{v_1, v_2,\dots, v_n\}$ — набор всех уязвимостей системы;

$\pi(v_k)$ — вероятность успешной эксплуатации уязвимости v_k.

Уязвимость v_k считается наиболее критичным компонентом тогда и только тогда, когда не существует такой уязвимости v_j (1J что $C_j > C_k$.

Авторы не учитывают зависимости между сервисами системы, финансовые потери в случае успешной реализации атак и затраты на внедрение контрмер.

В [140] байесовский граф атак определяется как кортеж:

$$BAG = (S,\tau,\varepsilon,\mathcal{P}),$$

где $S = N_{internal} \cup N_{external} \cup N_{terminal}$ — набор атрибутов (случайных переменных Бернулли, отображающих состояние экземпляра шаблона атрибута, соответствующего уязвимостям или небезопасным свойствам системы). Атрибут S имеет два свойства: состояние (TRUE или FALSE) и вероятность $Pr(S)$. Состояние TRUE (S=1) обозначает скомпрометированное состояние атрибута, $Pr(S)$ — вероятность того, что атрибут находится в состоянии $S = 1$, $\Pr(\neg S) = 1 - \Pr(S)$ — вероятность того, что атрибут находится в состоянии S=0;

$N_{internal}$ — набор атрибутов $S_j \in S$, для которых $\exists a_1, a_2 \in A | [S_j = pre(a_i)$ и $S_j = post(a_2)]$;

$N_{external}$ — набор атрибутов $S_i \in S$, для которых $\nexists a \in A | S_i = post(a)$;

$N_{terminal}$ — набор атрибутов $S_k \in S$, для которых $\nexists a \in A | S_k = pre(a)$;

$a \in A$ — набор атомарных атак, определенных на S;

$a: S_{pre} \rightarrow S_{post}$, где $S_{pre}, S_{post} \in S$, для которой существует условная зависимость $A: S \times S \rightarrow [0,1]$, называется атомарной атакой, если: (1) $S_{pre} \neq S_{post}$; (2) если $S_{pre} = 1$, $S_{post} = 1$ с вероятностью $A(S_{pre}, S_{post}) > 0$; (3) $\nexists S_1, \ldots, S_j \in S - \{S_{pre}, S_{post}\}$, таких что $A(S_{pre}, S_1) > 0$, $A(S_1, S_2) > 0$, ..., $A(S_j, S_{post}) > 0$. Атомарная атака связана с использованием уязвимости e с вероятностью $Pr(e)$, которая позволяет атакующему перейти из состояния S_{pre} в S_{post}, поэтому $A(S_{pre}, S_{post}) = Pr(e)$;

$\tau \in S \times S$ — набор связей; упорядоченная пара $(S_{pre}, S_{post}) \in \tau$, если $S_{pre} \rightarrow S_{post} \in A$. Для $S_i \in S$ набор $Pa[S_i] = \{S_i \in S | (S_j, S_i) \in \tau$ называется родительским набором;

ε — набор декомпозиционных кортежей вида $\langle S_j, d_j \rangle$, определенных для всех $S_j \in N_{internal} \cup N_{terminal}$ и , $d_j \in \{AND, OR\}$, d_j = OR, если $S_j = 1 \leftrightarrow \exists S_i \in Pa[S_j] | S_i = 1$, d_j =AND, если $S_j = 1 \leftrightarrow \forall S_i \in Pa[S_j] | S_i = 1$;

$\mathcal{P}$ — набор дискретных функций распределения условных вероятностей. Каждый атрибут $S_j \in N_{internal} \cup N_{terminal}$ имеет дискретное локальное распределение условной вероятности (LCDP), отображающее значения вероятностей, определяющие шансы компрометации узла, учитывая различные комбинации состояний его предков $\Pr(S_j | Pa[S_j])$. LCDP определяется следующим образом [133]. Если d_j = AND:

$$\Pr(S_j | Pa[S_j]) = \begin{cases} 0, & \exists S_i \in Pa[S_j] | S_i = 0, \\ \Pr(\cap_{S_i=1} e_i), & \text{иначе} \end{cases}$$

где $\Pr(\cap_{S_i=1} e_i) = \prod_{s_i=1} e_i$.

Если d_j =OR:

$$\Pr(S_j | Pa[S_j]) = \begin{cases} 0, & \forall S_i \in Pa[S_j] | S_i = 0, \\ \Pr(\cup_{S_i=1} e_i), & \text{иначе} \end{cases}$$

где $\Pr(\cup_{S_i=1} e_i) = 1 - \prod_{s_i=1} [1 - \Pr(e_i)]$.

В данной работе предложена методика определения вероятности атаки на основе байесовского графа атак. Методика включает следующие шаги:

Определение априорных вероятностей, в том числе:

a) определение вероятности осуществления угрозы $\Pr(S_i)$ для всех $S_i \in N_{external}$;

b) определение вероятности эксплуатации уязвимостей (LCDP) на основе формулы $\Pr(e_i) = 2 \times B\_AV \times B\_AC \times B\_AU$, где B_AV — вектор доступа CVSS, B_AC — сложность доступа CVSS, B_AU — аутентификация CVSS;

c) определение безусловных вероятностей $\Pr(S_j)$ для всех $S_j \in N_{internal} \cup N_{terminal}$ на основе формулы совместного распределения вероятностей: для набора состояний $S = \{S_1, \dots, S_n\}$, $\Pr(S_1, \dots, S_n) = \prod_{i=1}^{n} \Pr(S_i | Pa[S_i])$.

Вычисление апостериорных вероятностей в случае поступления событий безопасности (динамическая оценка риска) на основе теоремы Байеса: пусть $S = \{S_1, \dots, S_n\}$ — набор атрибутов BAG, $E = \{S'_1, \dots, S'_m\} \subset S$ — набор атрибутов, на которых были зафиксированы независимые события, то есть $S' = 1$ для всех $S'_j \in E$, $S_j \in S/E$ — атрибут, для которого необходимо определить апостериорную вероятность $\Pr(S_j|E)$, тогда $\Pr(S_j|E) = \frac{\Pr(E|S_j) \times \Pr(S_j)}{\Pr(e)}$, где $\Pr(E|S_j) = \prod_i \Pr(S'_i|S_j)$ — условная вероятность $S'_1, \dots, S'_m$ с учетом S_j, $\Pr(E) = \prod_i \Pr(S'_i)$, $\Pr(S_j)$ — априорные безусловные вероятности.

Достоинства подхода:

- возможность динамической оценки риска на основе поступающих событий.

Недостатки подхода:

- ручное определение вероятности начала атаки, без учета уровня навыков нарушителя;
- отсутствие учета распространения ущерба через зависимости сервисов.

В работе [140] применяются формулы определения условных вероятностей, учитывающие различные комбинации состояний предков узла, выведенные в [133]. В работе [133] также используется байесовский граф атак, на основе которого определяется общий уровень защищенности сети. Хотя методика позволяет делать выводы о защищенности на основе методов формального вывода, авторы не разрабатывают показателей защищенности. Кроме того, предлагаемая методика не автоматизирована.

Рассмотренные работы показали, что графы атак позволяют определить вероятность успешной реализации того или иного атакующего действия в сети, а также выявлять наиболее критичные точки системы.

Методики на основе графов зависимостей сервисов. С увеличением роли информационных технологий в работе организаций появилась необходимость интеграции бизнес-функций организации и информационных технологий. Это привело к развитию концепции сервис-ориентированных архитектур (СОА), которая уже достаточно давно внедряется такими крупными ИТ-компаниями, как Microsoft и IBM. В СОА процессы реализуются с помощью сервисов. С точки зрения сервис-ориентированных архитектур, сервис — это видимый ресурс, выполняющий повторяющуюся задачу и описанный внешней инструкцией [170]. Согласно другому определению, сервис — это ресурс, предоставляющий возможность выполнения задач, формирующих необходимую функциональность с точки зрения поставщиков и потребителей услуг [171]. То есть сервисы ориентированы на бизнес. Зависимости между сервисами позволяют определить обоснованность вло-

жений в ИТ за счет определения связи между бизнес-целями и технологиями. На рисунке 3.3 показана эта связь через взаимосвязи между следующими элементами [170]:

- эксплуатационные системы — существующие ИТ-решения. Данный уровень показывает ценность и важность вложений в ИТ для СОА;
- сервисные компоненты — компоненты, которые реализуют сервисы. Они могут использовать одно или более приложений с уровня эксплуатационных систем;
- сервисы — это сервисы, имеющиеся в системе;
- бизнес-процесс — это операционные программы, создающие бизнес-процессы в виде хореографий сервисов (хореография отражает обмен сообщениями, правила взаимодействий и соглашения между сервисами);
- пользователи — это каналы для доступа к бизнес-процессам, сервисам и приложениям.

Из-за сложности взаимосвязей между сервисами в сервис-ориентированных архитектурах не всегда просто определить, как именно та или иная уязвимость повлияет на деятельность организации. Ответом на эту проблему стали подходы к определению распространения ущерба в информационной структуре организации на основе графов зависимостей сервисов [144–147]. Граф зависимостей сервисов представляет собой множество сервисов компьютерной системы или сети, связанных между собой в соответствии с тем, как свойства безопасности одного сервиса зависят от свойств безопасности другого.

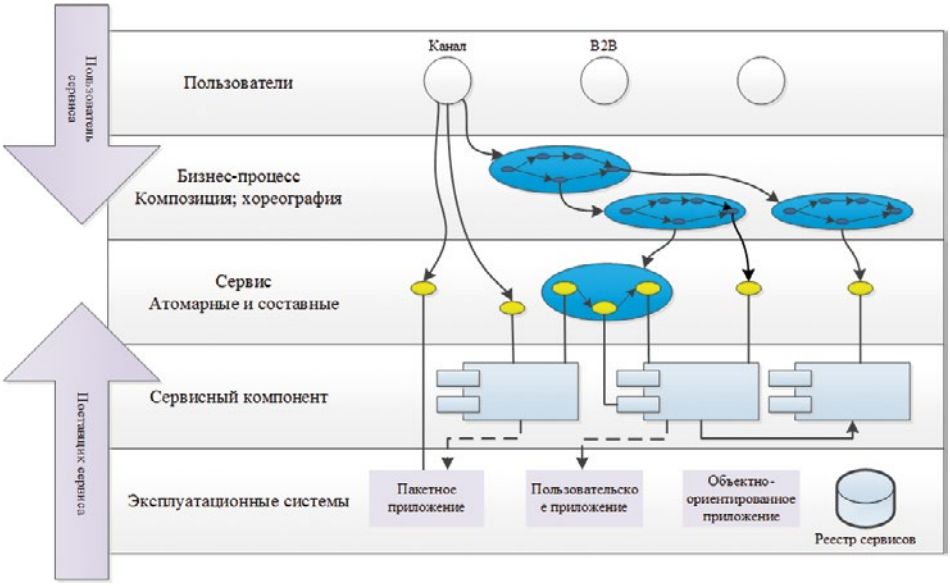


Рис. 3.3. Стек СОА-решения [170]

В [144] ресурсы (сервисы) структурируются во взвешенные деревья, где веса обозначают степень зависимости работоспособности ресурса от доступности другого ресурса.

В [145] была предложена карта системы, представляющая собой направленный граф зависимостей. Узлы данного графа соответствуют приоритетным ресурсам системы (ресурсам с ненулевой стоимостью) и ресурсам, необходимым для корректного функционирования приоритетных ресурсов. Для каждого узла определена стоимость, а также группа к которой относится соответствующий ресурс (каждой группе соответствует определенный набор контрмер).

В [135] сделан шаг в сторону объединения графов атак и графов зависимостей сервисов. В работе используется направленный граф, называемый графом вторжений, в котором узлы соответствуют целям нарушителя (в компрометации сервисов), а дуги обозначают пред- и постусловия достижения целей. Дуги могут быть связаны отношениями: И, ИЛИ, Кворум (для достижения цели должно быть достигнуто определенное количество подцелей). Граф строится на основе уязвимостей и сервисов системы. Сервисы отображаются в виде направленного графа, дуги которого показывают возможности распространения вторжения через отношения между сервисами. При получении сообщения об эксплуатации уязвимости выделяется узел на графе вторжений, соответствующий данной уязвимости с точки зрения затронутого сервиса и уязвимостей и сервисов, которые должны быть скомпрометированы для эксплуатации данной уязвимости, и формируется подграф соответствующей атаки в виде сети Петри.

В [147] применяется граф зависимостей между доступностью ресурсов.

В [146] авторы используют модель зависимостей сервисов в виде направленного графа $G=(Sr, Ar)$, где Sr — набор ресурсов r_i системы (сервисы и пользователи), $Ar \subset Sr \times Sr$– множество зависимостей между ресурсами, определяется как $\{(r_i, r_j) \in Ar \leftrightarrow r_j \in Ant(r_i)\}$, $i, j \in [1, N]$, N — количество ресурсов в системе, r_j — зависимый сервис, $Ant(r_i)$ набор прямых предков r_i, т.е. ресурсов, необходимых для корректного функционирования r_i. Автор выделяет два типа зависимостей: структурные (зависимости между сервисами различных уровней модели ISO/OSI) и функциональные (между разными сервисами одного уровня). Пример структурных зависимостей: зависимость между веб-приложением, сервером JBoss и портом tcp/443 на рисунке 3.4. Пример функциональных зависимостей: зависимость между веб-приложением и аутентификацией на рисунке 3.5.

В [148] в модель добавлено понятие привилегий сервиса в отношении актива. Привилегии передаются между зависимыми сервисами через отношения доверия. Например, если существует сервис B с привилегиями R ($B.R$) и зависимый сервис A с привилегиями Pr или учетной записью Cr ($A.Pr$ или $A.Cr$), то сервис B может передать свои привилегии R сервису A, если доверяет ему: $A \xrightarrow{(A.Cr, A.Pr)} B.R$. При этом ущерб конфиденциальности и доступности является параметром актива, а ущерб доступности связан также с сущностью, использующей актив, и является параметром привилегий.

Учет распространения ущерба через зависимости сервисов позволит отрегулировать затраты на безопасность, чтобы они не превысили возможный ущерб, не упустить важные уязвимости, которые могут привести к серьезным последствиям, а также обосновать затраты на безопасность.

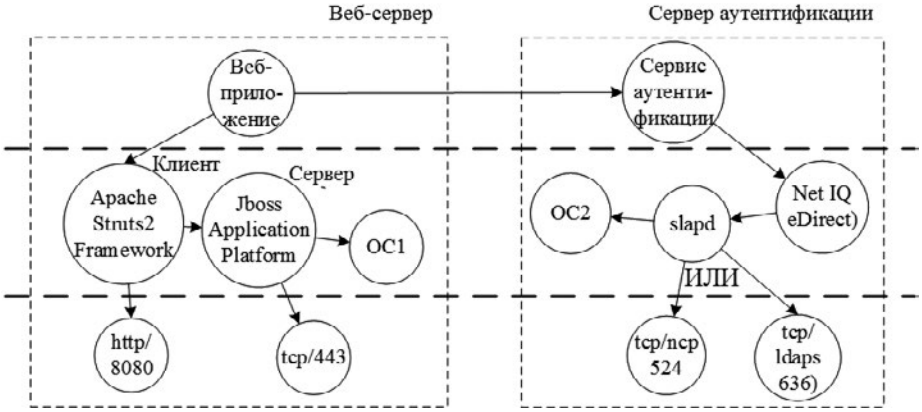


Рис. 3.4. Пример функциональных и структурных зависимостей [146]

Модель, предложенная в [146], позволяет учесть распространение ущерба для всех трех свойств безопасности (конфиденциальности, целостности и доступности). Тем не менее она не позволяет учитывать вероятность нанесения ущерба ресурсу, и, таким образом, подход на основе такой модели актуален только для проходящей атаки. Данную проблему можно решить за счет формирования связи между моделями атак и зависимостей сервисов.

Для определения ущерба, наносимого атакующими действиями, в рассмотренных выше работах предлагаются различные показатели защищенности.

В [144] для определения побочного ущерба при реагировании предлагается показатель *penalty cost* $p(r)$, отражающий стоимость снижения производительности сервиса r в результате потери его доступности: $p(r)=cr(r)\cdot penalty$. Где *penalty* — значение, отражающее значимость сервиса (определяется пользователем); $cr(r)$ — уровень снижения производительности: $cr(r)=1-c(r)$; $c(r)$ — итоговая производительность сервиса r, которая определяется на основе дерева зависимостей между ресурсами.

В [145] показатель ущерба от атаки рассчитывается на основе карты системы (граф зависимостей, объединяющий приоритетные ресурсы), иерархии ресурсов (группировка ресурсов по типам с выделением контрмер для каждого типа) и стоимостной модели (в которой ресурсам назначаются стоимости) как сумма стоимостей узлов системной карты, на которые атакующее действие повлияло негативно.

В работе [147] получила дальнейшее развитие концепция производительности ресурсов, предложенная [144]. Стоимость ущерба, нанесенного в результате проведения атаки, рассчитывается как значение доступности ресурсов после проведения атаки (в процентном отношении). При этом доступность ресурсов зависит от внутренней доступности ресурса и доступности ресурсов, необходимых для его функционирования.

В [146] предлагается подход к определению ущерба, наносимого системе в результате успешной реализации атаки, с учетом распространения негативного влияния через зависимости сервисов. На рисунке 3.5 представлена схема предлагаемого подхода к определению ущерба.

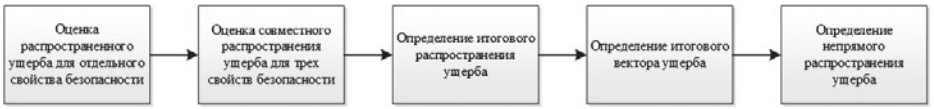


Рис. 3.5. Подход к определению уровня распространяемого ущерба

На первом шаге — оценка распространенного ущерба для отдельного свойства безопасности (конфиденциальности, целостности или доступности) — вычисляется ущерб $DV_{r_i,r_j}[k,l]_{k,l\in\{1,2,3\}}$, наносимый k-ому свойству безопасности ресурса r_i ($k = 1$ — конфиденциальности, $k = 2$ — целостности и $k = 3$ — доступности) через зависимость $(r_i, r_j) \in Ar$ при нарушении свойства безопасности l ресурса r_j. Ущерб может распространяться линейно (если зависимость (r_i, r_j) имеет вес $w_{ri,rj} \leq 1$), тогда $DV_{r_i,r_j}[k,l] = w_{r_i,r_j} \times V_{r_j}[l]$, где $V_{r_j}[l] \in [0,1]$ — вектор-столбец, отображающий степень нарушения l-го свойства безопасности ресурса r_j, $w_{r_i,r_j}[k,l]$ – вес зависимости k-ого свойства безопасности ресурса r_j от свойства безопасности l ресурса r_j. V_r зависит от внутреннего состояния ресурса r (IV_r) и от влияния некорректной работы зависимых ресурсов (DV_r). Другой тип распространения ущерба — усиливающее распространение (зависимость (r_i, r_j) имеет вес $w_{ri,rj} \geq 1$), тогда $DV_{r_i,r_j}[k,l] = \begin{cases} w_{r_i,r_j} \times V_{r_j}[l], & \text{если } V_{r_j}[l] \leq \frac{1}{w_{r_i,r_j}[k,l]} \\ 1, & \text{если } V_{r_j}[l] > \frac{1}{w_{r_i,r_j}[k,l]} \end{cases}$. Последний тип распространения ущерба — распространение с ограничением. Свойства безопасности ресурса r_i не подвергаются ущербу, пока $V_{r_i}[l]$ не превышает определенный порог $w_{r_i,r_j}[k,l] < 1$. Тогда $DV_{r_i r_j}[k,l] = \begin{cases} 0, & \text{если } V_{r_j}[l] \leq w_{r_i,r_j}[k,l] \\ 1, & \text{если } V_{r_j}[l] > w_{r_i,r_j}[k,l] \end{cases}$.

На втором шаге вычисляется совместное распространение ущерба для трех свойств безопасности ресурса на основе матрицы зависимости свойств безопасности $k \in \{1,2,3\}$ ресурса r_i от свойств безопасности $l \in \{1,2,3\}$ ресурса r_j: $W_{r_i r_j} = \begin{pmatrix} w_{r_i,r_j}[1,1] & w_{r_i,r_j}[1,2] & w_{r_i,r_j}[1,3] \\ w_{r_i,r_j}[2,1] & w_{r_i,r_j}[2,2] & w_{r_i,r_j}[2,3] \\ w_{r_i,r_j}[3,1] & w_{r_i,r_j}[3,2] & w_{r_i,r_j}[3,3] \end{pmatrix}$. Итоговый ущерб для k-го свойства безопасности ресурса определяется как $DV_{r_i,r_j}[k]_{k\in\{1,2,3\}} = max_l(w_{r_i,r_j}[k,l] \times \times V_{r_j}[l])_{l\in\{1,2,3\}}$.

На третьем шаге вычисляется итоговое распространение ущерба DV_r для ресурса r с учетом конъюнктивных (работоспособность сервиса зависит от корректной работы нескольких сервисов) и дизъюнктивных (для работоспособности сервиса необходима корректная работа одного из нескольких сервисов) зависимостей графа. Для конъюнктивных зависимостей DV_r определяется как максимальный ущерб из всех $\{DV_{rr_n}\}_{r_n\in Anr(r)}$: $DV_r = = \begin{pmatrix} max_n\{DV_{rr_n}[1]\}_{n=1..m} \\ max_n\{DV_{rr_n}[2]\}_{n=1..m} \\ max_n\{DV_{rr_n}[3]\}_{n=1..m} \end{pmatrix}$, где m — число ресурсов предков.

Для дизъюнктивных зависимостей ущерб доступности ресурса определяется как минимальный ущерб доступности для ресурсов предков,

а ущерб целостности и конфиденциальности определяется как среднее значение ущерба по всем дизъюнктивным зависимостям с учетом ограничения распространения ущерба целостности и конфиденциальности ущербом доступности: $DV_r = \begin{pmatrix} \frac{\sum_{k=1}^{m}(1-V_{r_k}[3])\times DV_{r_k}[1]}{\sum_{k=1}^{m}(1-V_{r_k}[3])} \\ \frac{\sum_{k=1}^{m}(1-V_{r_k}[3])\times DV_{r_k}[2]}{\sum_{k=1}^{m}(1-V_{r_k}[3])} \\ min_k\{DV_{rr_n}[3]\}_{k=1..m} \end{pmatrix}$.

На четвертом шаге определяется итоговый вектор ущерба V_r ресурса r на основе векторов IV_r и DV_r: $V_r = \begin{pmatrix} \max(IV_r[1], (1-IV_r[3])\times DV_r[1]) \\ \max(IV_r[2], (1-IV_r[3])\times DV_r[2]) \\ IV_r[3]+DV_r[3]-IV_r[3]\times DV_r[3] \end{pmatrix}$.

И, наконец, на пятом шаге на основе алгоритма Форда-Фалкерсона определяется непрямое распространение ущерба по графу зависимостей сервисов. Работа алгоритма начинается с узлов, для которых изменилось значение внутреннего ущерба IV_r. Для определения ущерба, нанесенного определенному узлу, значение V_r умножается на вектор-столбец критичности. В случае если ущерб нанесен нескольким ресурсам, результирующий ущерб определяется суммой ущерба по всем ресурсам.

Как говорилось выше, данный подход имеет преимущество перед другими подходами на основе моделей зависимостей сервисов, так как позволяет определить не только распространение ущерба доступности, но и распространение ущерба целостности и конфиденциальности сервисов.

Рассмотренные работы показали, что при определении уровня защищенности системы важно учитывать распространение ущерба через зависимости сервисов с точки зрения конфиденциальности, целостности и доступности. Для этого целесообразно применять графы зависимостей сервисов и определяемый на их основе показатель распространенного ущерба.

Стоимостные методики оценивания защищенности

RiskWatch. Компания RiskWatch разработала собственную методику анализа рисков [172]. Методика RiskWatch в качестве критерия для оценки и управления рисками использует ожидаемые годовые потери (Annual Loss Expectancy, ALE) и оценку возврата инвестиций (Return on Investment, ROI). RiskWatch ориентирована на точную количественную оценку соотношения потерь от угроз безопасности и затрат на создание системы защиты. В основе продукта RiskWatch находится методика анализа рисков, которая состоит из четырех этапов:

• определение предмета исследования (защищаемых ресурсов, потерь, угроз, уязвимостей и мер защиты) экспертами на основе заполнения соответствующих таблиц;

• ввод данных, описывающих конкретные характеристики системы (ценность ресурсов, уязвимости ресурсов, степень уязвимости, частота возникновения угроз, потери и классы инцидентов), вручную или с использованием инструментальных средств;

• количественная оценка риска (расчет рисков и выбор мер обеспечения безопасности);
• генерация отчетов.

Риск оценивается на основе ожидаемых потерь за год:

$$ALE = AssetValue \cdot ExposureFactor \cdot Frequency,$$

где *ALE* — ожидаемые потери за год для одного конкретного актива от реализации одной угрозы;

AssetValue — стоимость рассматриваемого актива;

ExposureFactor — коэффициент, который определяет какая часть от стоимости актива подвергается риску (в процентах);

Frequency — частота возникновения нежелательного события в форме LAFE или SAFE. LAFE и SAFE — это определенные американским институтом стандартов NIST оценки. LAFE (Local Annual Frequency Estimate) показывает, сколько раз в год в среднем данная угроза реализуется в данном месте (например, в городе). SAFE (Standard Annual Frequency Estimate) показывает, сколько раз в год в среднем данная угроза реализуется в данной части мира (например, в Северной Америке).

Общий риск для компьютерной системы оценивается как сумма всех частных значений.

Эффект от внедрения средств защиты количественно описывается с помощью показателя возврата инвестиций *ROI* (Return on Investment):

$$ROI = \sum_i NVP(Benefits_i) - \sum_j NVP(Costs_j),$$

где *NVP* (Net Present Value) — функция определения чистой текущей стоимости;

$Benefits_i$ — оценка ожидаемого снижения потерь в результате внедрение *i*-ой защитной меры;

$Costs_j$ — затраты на внедрение и поддержку *j*-ой защитной меры.

Достоинства методики:

методика позволяет оценить не только те риски, которые существуют у предприятия на данный момент, но и ту выгоду, которую может принести внедрение физических, технических, программных и прочих средств и механизмов защиты;

итоговые отчеты и графики дают материал, достаточный для принятия решений об изменении системы обеспечения безопасности предприятия.

Недостаток методики:

полученный ущерб будет в итоге выше, чем реальный ущерб, т.к. на один и тот же актив может быть реально направлено сразу несколько угроз, что приведет к тому, что суммарный ущерб, подсчитанный по угрозам, будет неадекватен реальному, подсчитанному по активам.

Методики, основанные на интегральном показателе поверхности атаки

Методика оценивания защищенности Microsoft. Компания Microsoft предложила меру оценки уровня защищенности программного обеспечения системы — *относительный коэффициент поверхности атаки* (Relative Attack Surface Quotient, RASQ) и соответствующий инструмент анализа поверхности атаки [173]. Предложенный коэффициент основан на концепции поверхности

атаки, позднее развитой в работах [174–177]. Он позволяет квантифицировать относительную подверженность атаке для ИТ-активов предприятия.

Поверхность атаки определяется на основе тройки ресурсов, которые могут использоваться при проведении атаки: (1) методов, которые могут получать и отправлять данные; (2) каналов, которые используются для передачи данных; (3) данных.

Для определения поверхности атаки может использоваться следующая формула:

$$Attack\_Surface = (\sum_{m \in M} DER_m(m), \sum_{c \in C} DER_c(c), \sum_{i \in I} DER_i(i)),$$

где $DER_m = \frac{privilege}{access\_rights}$ — оценивается в терминах привилегий на точках входа/выхода;

$DER_c = \frac{protocol}{access\_rights}$ — для каналов оценивается в терминах ограничений, накладываемых на данные протоколом;

$DER_i = \frac{data\_type}{access\_rights}$ — оценивается с точки зрения подозрительных данных.

Вклад каждого ресурса в значение показателя «поверхность атаки» определяется при помощи показателя «отношение потенциал разрушений-усилия» (*Damage Potential-Effort Ratio*). Меньшие усилия (предусловия) и больший потенциал разрушений (постусловия) ведут к большему значению показателя.

Показатель «поверхность атаки» напрямую связан с риском — чем больше поверхность атаки, тем больше риск компрометации системы.

Относительный коэффициент поверхности атаки определяется сложением значений эффективной поверхности атаки для всех возможных векторов атаки. Вектора атаки — это определенные характеристики ОС, которые могут положительно или отрицательно влиять на защищенность продукта. Значение эффективной поверхности атаки определяется на основе количества поверхностей атаки внутри вектора атаки (таких, как запущенные по умолчанию сетевые сервисы, плохо защищенные учетные записи и файлы и т.п.) и риска компрометации для вектора атаки относительно определенной угрозы компрометации. Риск компрометации определяется на основе уязвимости поверхности атаки и ее привлекательности для потенциальных атакующих на шкале от 0 (нет угроз) до 1 (максимальная угроза).

Методики, основанные на экспертных знаниях

CRAMM. Методика CRAMM [178] была разработана Центральным компьютерным и телекоммуникационным агентством (Central Computer and Telecommunications Agency, CCTA) Великобритании и является качественно-количественной методикой оценки рисков. Методика включает три этапа: выявление и оценка активов (это могут быть данные, ПО или аппаратные ресурсы); выявление угроз и уязвимостей и оценка рисков; выявление и приоритизация защитных мер.

Ценность активов определяется в денежных единицах. Для оценки возможного ущерба используется шкала со значениями от 1 до 10. При низкой оценке (3 балла и ниже) считается, что рассматриваемая система требует базового уровня защиты, и вторая стадия исследования пропускается.

Оценка уровней угроз и уязвимостей проводится на основе экспертных знаний с помощью списков вопросов. Выделяются следующие уровни угроз: очень высокий, высокий, средний, низкий и очень низкий. Возможные уровни уязвимости: высокий, средний и низкий.

Уровни рисков рассчитываются на основе ожидаемых годовых потерь по шкале от 1 до 7. Ожидаемые годовые потери определяются на основе оценок стоимости активов, уровня угрозы и уровня уязвимости.

На основе уровней рисков генерируются варианты защитных мер.

Программные средства оценивания защищенности

RiskWatch. Семейство программных средств RiskWatch [172] реализует методику RiskWatch, описанную выше. В семейство RiskWatch входят:

RiskWatch for Physical Security — средство анализа физической защиты информационной системы (ИС);

RiskWatch for Information Systems — средство анализа информационных рисков;

HIPAA-WATCH for Healthcare Industry — средство оценки соответствия требованиям стандарта HIPAA (US Healthcare Insurance Portability and Accountability Act), актуального в основном для медицинских учреждений, работающих на территории США;

RiskWatch RW17799 for ISO 17799 — средство оценки соответствия ИС требованиям международного стандарта ISO 17799.

Инструмент TVA (Topological Analysis of Network Attack Vulnerability, Топологический анализ сетевых уязвимостей). Инструмент TVA [179–181] позволяет генерировать графы зависимостей между эксплоитами (на основе их пред- и постусловий). Пути атак формируются обходом данного графа. Используемый алгоритм имеет полиномиальную сложность и подразумевает монотонность атак [130].

Система NETSPA (A Network Security Planning Architecture, Архитектура планирования сетевой безопасности). Система NETSPA [165, 182] формирует граф атак, используя правила брандмауэра (для определения достижимости хостов) и известные уязвимости анализируемой сети. Система позволяет сформировать граф с учетом одного или нескольких исходных положений злоумышленника. Используемый алгоритм имеет сложность $O(n\log n)$, где n — количество хостов анализируемой сети. Риск оценивается как количество активов, которые могут быть скомпрометированы злоумышленником [130].

Система MULVAL (Multihost, multistage, Vulnerability Analysis, Многохостовый, многоступенчатый анализ уязвимостей). Система MULVAL [168, 183] представляет собой систему анализа защищенности на основе Datalog, использующую конфигурационную информацию и информацию об уязвимостях анализируемой сети. Механизм вывода позволяет выявить связи между различными компонентами сети, представленными в виде фактов Datalog. Используемый алгоритм имеет сложность $O(n^2)$, где n — количество хостов анализируемой сети [130].

Программное обеспечение SecurITree. ПО SecurITree от компании Amenaza [184] предназначено для построения и анализа деревьев атак. Позволяет формировать возможные сценарии атак с учетом целей атакую-

щего и необходимых предусловий атаки. Кроме того, данное средство позволяет анализировать риски безопасности с использованием следующих показателей:

- стоимость сценария атаки;
- ресурсы атакующего;
- стоимость сценария атаки для атакующего, рассчитываемая на основе стоимости сценария атаки и ресурсов атакующего;
- выигрыш атакующего;
- вероятность сценария атаки (относительная вероятность каждой реализации сценария атаки), рассчитываемая на основе стоимости сценария атаки для атакующего и выигрыша атакующего;
- количество зафиксированных реализаций сценария атаки за период;
- совокупная вероятность сценария атаки, рассчитываемая на основе количества зафиксированных реализаций сценария атаки за период и вероятности сценария атаки;
- потери от реализации сценария атаки;
- ожидаемые потери за период, рассчитываемые на основе совокупной вероятности сценария атаки и потерь от реализации сценария атаки;
- ущерб от реализации сценария атаки;
- относительный риск, рассчитываемый на основе вероятности сценария атаки и ущерба;
- совокупный риск, рассчитываемый на основе совокупной вероятности сценария атаки и ущерба.

Выводы по главе 3

В области оценивания защищенности с использованием показателя риска существует достаточно большое количество методик как качественного, так и количественного оценивания.

К достоинствам методик качественного оценивания относятся:

- низкие временные затраты (дни, а не недели или месяцы);
- эффективность по стоимости.

К достоинствам методик количественного оценивания относятся:

- возможность количественно оценить и сравнить риски, которые существуют у предприятия на данный момент, и ту выгоду, которую может принести внедрение физических, технических, программных и прочих средств и механизмов защиты;
- итоговые отчеты и графики дают материал, достаточный для принятия решений об изменении системы обеспечения безопасности предприятия.

При этом и те и другие обладают рядом недостатков. Так, методики качественного оценивания не позволяют получить или разработать показатели защищенности для оценивания вероятности угрозы или годовых потерь. Кроме того, такие методики субъективны, так как основаны на использовании экспертных знаний.

Методики количественного оценивания требуют чрезмерных временных и материальных затрат, документация по рискам становится слишком объемной для практического применения.

Кроме того, в современных условиях к методикам оценивания защищенности предъявляются дополнительные требования. Количество и сложность атак на ИТ-инфраструктуры предприятий растет. При этом даже небольшое время нарушения корректного функционирования ИТ-систем может привести к серьезным финансовым потерям, а своевременное обнаружение атакующего и принятие соответствующих контрмер может их предотвратить. Однако, из-за сложности и размеров современных ИТ-инфраструктур, процесс оценивания защищенности может занять значительное время. Это еще более усложняется тем, что для обоснованного выбора контрмер, которые позволят минимизировать ущерб от кибератаки и не нанесут дополнительного вреда, требуются объективные количественные показатели. Таким образом, с одной стороны, для своевременного и эффективного реагирования на компьютерные атаки необходимо провести полноценное количественное оценивание защищенности и рисков, с другой стороны, для этого требуются серьезные временные и материальные ресурсы.

Глава 4. Методики и средства выбора контрмер

Средства технической защиты информации и средства обеспечения кибербезопасности информационных технологий могут выбираться на этапе проектирования компьютерной системы или сети (например, в соответствии с ПЗ ФСТЭК). При этом компьютерная система или сеть все еще может подвергаться успешным кибератакам, например, с использованием неизвестных ранее уязвимостей. Для управления кибербезопасностью на этапе эксплуатации компьютерной системы или сети выделяются реактивный и проактивный подходы к выбору контрмер. Реактивный подход подразумевает реагирование на уже совершившуюся кибератаку, в то время как проактивный подход подразумевает прогнозирование развития кибератаки и принятие мер до того, как ее выполнение нанесет серьезный ущерб.

4.1. Методики выбора контрмер

Вопросы выбора контрмер рассматриваются многими исследователями [135, 138, 140, 144, 145, 147–149, 164, 165, 185–189].

В ряде работ рассматриваются методики на основе теории игр [186, 188]. В других — методики принятия решений на основе логического вывода на графах атак [164, 165, 185]. В некоторых работах предлагается динамический выбор защитных мер на основе байесовских графов атак [138, 140]. В [149, 190] описываются методики определения общего уровня защищенности системы, который затем используется для сравнения уровня защищенности до и после внедрения защитных мер. В [135, 144, 145, 147, 148, 187, 189] рассматривается определение ожидаемых потерь и эффективности контрмер. В [191, 192] используются экономические индексы для оценивания возможных потерь и эффективности контрмер. Преимущество таких методов состоит в установлении прямой зависимости между финансовыми затратами организации и потерями от атак. Наиболее интересные из перечисленных подходов рассмотрены ниже.

В [188] рассматривается анализ сетевой безопасности с применением теории игр. Вводятся понятия стоимости и выигрыша, что позволяет учесть финансовый аспект при анализе сетевой безопасности (стоимость проведения атаки, ущерб от ее проведения и средства, необходимые для ее предотвращения). Результатом работы предлагаемого подхода является оптимальная стратегия — то есть наилучшая последовательность действий, которую администратор или нарушитель может применить для достижения своих целей (например, найти наиболее эффективную стратегию установления патчей в терминах стоимости и временных затрат). Преимущество методов на основе теории игр — возможность учета временного аспекта.

Игра (стратегия) описывается как последовательность состояний и действий. Для описания состояния используется граф зависимостей между сервисами и файлами системы и ряд свойств для каждого узла (уязвимость, скомпрометированность и др.). Изменение состояния происходит при применении действий (правил), включающих предусловие, время выполнения,

стоимость, действие, выигрыш (если есть) и игрока, выполняющего действие. Для выбора стратегии, минимальной по стоимости и времени, авторы предлагают минимизировать сумму затрат и времени, необходимых для выполнения всех действий в стратегии.

В [186] предлагается методика оценивания рисков, основанная на теории игр. Авторы вводят теоретико-игровую модель защиты от атак (Game Theoretical Attack-Defense Model, GTADM) и иерархическую модель вычисления рисков (Hierarchical Risk Computing Model, HRCM).

Авторы выделяют ряд сущностей для определения процесса оценивания рисков: СЕТЬ (объект оценки сетевых рисков), ЗНАНИЕ (отражает знание атака-защита, используемое в процессе оценки, которое включает параметры стоимости и выигрыша), АКТИВ (ценность полезных файлов, сервисов и т.п. Она отображается как функция от конфиденциальности, целостности и доступности), УЯЗВИМОСТЬ (дефекты или слабые места, которые могут использоваться злоумышленниками и привести к разрушениям в системе), УГРОЗА (потенциальные атаки извне), УПРАВЛЕНИЕ (защитные меры, применяемые в сети), УЩЕРБ (последствия, если потенциальные атаки произойдут в текущем состоянии системы), ВЕРОЯТНОСТЬ (вероятность угроз), РИСК (потенциальные потери, вызываемые угрозами при использовании уязвимостей).

Кроме того, оценивание рисков включает ряд компонентов (действий), которые определяются как функции преобразования входных данных в выходные:

1. Определение активов e_1 (сбор информации об активах и ценности системы на основе конфигурации сети):

$$\varphi^{e_1}: NETWORK \longrightarrow ASSET.$$

2. Определение средств управления e_2 (информация о сети преобразовывается в информацию о средствах защиты):

$$\varphi^{e_2}: NETWORK \longrightarrow CONTROL.$$

3. Определение уязвимостей e_3 (сбор информации об уязвимостях из открытых баз данных и от средств обнаружения уязвимостей):

$$\varphi^{e_3}: NETWORK \times KNOWLEDGE \longrightarrow VULNERABILITY.$$

4. Определение угроз e_4 (включает определение угроз на основе знаний об уязвимостях и отношениях между уязвимостями и атаками и определение угроз, основанных на данных угрозах и отношениях доверия *THREAT'*):

$$\varphi^{e_4}: VULNERABILITY \times KNOWLEDGE \longrightarrow THREAT,$$

$$\varphi^{e_4}: THREAT \times TRUST \longrightarrow THREAT'.$$

5. Определение ущерба e_5 (определяется ущербом, вызванным атаками на конфиденциальность, целостность и доступность (Con_p, Int_p, $Ava_p \in [0,1]$)):

$$\varphi^{e_5}: THREAT' \times KNOWLEDGE \times DETECTION \times ASSET \longrightarrow IMPACT.$$

6. Определение вероятности e_6 (вероятность атак позволяет определить потенциальную стратегию злоумышленников):

$$\varphi^{e_6}: THREAT' \times KNOWLEDGE \times ASSET \longrightarrow PROBABILITY.$$

7. Вычисление рисков e_7 (риск каждой угрозы вычисляется на основе ожидаемых потерь и вероятности атак):

$$\varphi^{e_7}: IMPACT \times PROBABILITY \longrightarrow RISK.$$

Авторы определяют оценку рисков *S* на основе компонентов *E* и их отношений *R* следующим образом:

$$\left\{\begin{matrix} S ::= (E,R) \quad E ::= \{e_1,e_2,e_3,e_4,e_5,e_6,e_7\} \\ R ::= \{\langle e_1,e_5\rangle,\langle e_1,e_6\rangle,\langle e_2,e_5\rangle,\langle e_3,e_4\rangle,\langle e_4,e_5\rangle,\langle e_4,e_6\rangle,\langle e_2,e_4\rangle,\langle e_5,e_7\rangle,\langle e_6,e_7\rangle\} \end{matrix}\right\}.$$

Авторы предлагают трехуровневую структуру оценивания рисков. На уровне сбора данных формируются данные о статусе сети. На уровне обработки информации на основе данных о статусе сети определяются уязвимости системы, а на основе знаний об отношениях между уязвимостями и угрозами, и физических и логических соединениях между узлами, определяются все возможные атаки на текущую сеть. На уровне вычисления рисков строится модель GTADM для вычисления вероятностей угроз и ожидаемого ущерба в результате успешной реализации каждой угрозы и вычисляется статус рисков текущей системы на основе HRCM.

GTADM описывается как некооперативная статическая игра с ненулевой суммой и полной информацией. Игроками являются злоумышленник и защитник. Пространства стратегий: $\{att_i(o), \neg att_i(o)\}$ и $D(att_i(o))$, $\neg D(att_i(o))$; полезность для обеих сторон определяется как $Benefit_{ij}$-$Cost_{ij}$.

Выгода для защитника определяется на основе *S_Damage* (разрушения, в случае если методы защитника не сработали. Оно может быть определено как функция от степени разрушений в результате атаки на объект атаки и активов объекта) и *Restore* (определяет восстановление, когда атака обнаружена).

Стоимость для защитника определяется на основе *O_Cost* (стоимость анализа и развертывания мер защиты, которая может быть проигнорирована по сравнению с *R_Cost* и *A_Cost*), *R_Cost* (стоимость мер восстановления) и *A_Cost* (разрушения доступности системы, вызванные защитными мерами). *A_Cost* определяется как: $A\_Cost = -P_a \cdot Ava_v$, где $P_a \in [0,1]$.

Выгода для атакующего определяется на основе *Att_Benefit* (выигрыш злоумышленника, совпадающий численно с выигрышем защитника).

Стоимость для атакующего определяется на основе стоимости инструментария для атаки (*AC*) и наказания в случае обнаружения (*Punish*), где *AC* можно опустить.

На основе анализа выше строится платежная матрица GTADM (таблица 4.1), где *p* — плотность правильных обнаружений IDS, p^m — плотность ложных обнаружений.

Таблица 4.1

Платежная матрица GTADM [186]

	$D(att_i(o))$	$\neg D(att_i(o))$
$att_i(o)$	$\begin{pmatrix} \begin{pmatrix} S\_Damage(att_i(o)) \cdot (1-p) + (S\_Damage(att_i(o)) - Restore) \cdot p \\ -p \cdot Att\_Pun \end{pmatrix} \\ \begin{pmatrix} -S\_Damage(att_i(o)) \cdot (1-p) + (S\_Damage(att_i(o)) - Restore) \cdot p \\ -(R_{Cost} + P_a \cdot Ava_v) \cdot p \end{pmatrix} \end{pmatrix}$	$\begin{pmatrix} S\_Damage(att_i(o)) \\ -S\_Damage(att_i(o)) \end{pmatrix}$
$\neg att_i(o)$	$(0, -(R\_Cost + P_a \cdot Ava_v) \cdot p^m)$	(0,0)

Для вычисления вероятности того, что злоумышленник предпримет действие $att_i(o)$ авторы используют теорему существования равновесия Нэша:

$$P_{att_i(o)} = \frac{1}{1+\frac{p}{p^m}\cdot\frac{(Restore-R_{Cost}-P_a\cdot Ava_v)}{R_{Cost}+P_a\cdot Ava_v}},$$

где $P_{att_i(o)} \in [0,1]$.

В HRCM риски вычисляются снизу вверх: для угроз, затем для узлов и затем для сети.

Риск вычисляется по формуле:

$$Risk_{att_i(o)} = P_{att_i(o)} \cdot ED_{att_i(o)},$$

где $ED_{att_i(o)}$ — ожидаемые потери.

Значение риска узла — сумма значений рисков всех угроз узла. Риск системы — сумма рисков всех узлов.

В [165] предлагается подход к генерации графа атак с множеством предусловий, позволяющий выявить наиболее критичные уязвимости системы (дающие нарушителю наибольший доступ в системе). Подход реализован в виде программного средства NetSPA (NETwork Security and Planning Architecture). Основным преимуществом подхода является его применимость для оперативного построения графов для сетей большого масштаба. Подход включает сбор данных из открытых источников, автоматическое определение на их основе достижимости для хостов (то есть возможности данного хоста связаться с открытыми портами других хостов системы) и генерацию графа атак. Предусловиями атакующего действия является достижимость хоста и наличие на нем уязвимостей или ошибок конфигурации. Постусловиями является получение одного из четырех видов доступа на хосте: администраторский доступ; гостевой доступ; отказ в обслуживании и другое (потеря конфиденциальности и/или целостности). Достижимость хоста для нарушителя определяется наличием администраторского или гостевого доступа. Авторы формируют рекомендации по реализации защитных мер на основе показателей, определяющих, как много путей атаки будет заблокировано введением защитной меры.

Тем не менее в работе не описаны конкретные показатели и не предложено методик их вычисления и методик определения уровня защищенности.

В [164] рассматривается стратегия защиты в глубину на основе применения защитных мер по уровням графа атак. В отличие от [165] более подробно рассматривается применение инструмента NetSPA для усиления защиты. Для оценивания защищенности сети предлагается показатель *процент компрометации сети* (Network Compromise Percentage, NCP), который определяет процент хостов сети, на которых атакующий может получить права пользователя или администратора. Показатель принимает значения от 0 до 100%.

Для выбора защитных мер предлагается следующий алгоритм: граф обходится в ширину, и для каждого хоста создается список входящих дуг (соответствующих уязвимостям, которые могут быть использованы для компрометации хоста). Для этих дуг подбираются защитные меры. Кроме того, создаются группы хостов, для которых могут использоваться одни и те же

меры. Группировка осуществляется следующим образом: если множество потомков рассматриваемого узла пересекается со множеством потомков узла одной из групп, узел добавляется в данную группу (так как очевидно, что компрометация данного узла усиливает вероятность компрометации его потомков, а устранение уязвимостей, ведущих к этой компрометации, напротив, эту вероятность уменьшает). Каждая защитная мера устраняет дугу графа. Таким образом, для каждой защитной меры рассчитывается NCP после реализации меры, и вычисляется разница между предыдущим и новым значением NCP. Затем меры упорядочиваются в соответствии с максимальной разницей.

Тем не менее в работе не рассматриваются подробные алгоритмы вычисления уровня риска с учетом ущерба и вероятности атаки.

В [144] для определения побочного ущерба при реагировании предлагается показатель *penalty cost* $p(r)$, отражающий стоимость снижения производительности сервиса r в результате потери его доступности: $p(r)=cr(r)\cdot penalty$, где *penalty* — значение, отражающее значимость сервиса (определяется пользователем); $cr(r)$ — уровень снижения производительности: $cr(r)=1-c(r)$; $c(r)$ — итоговая производительность сервиса r, которая определяется на основе дерева зависимостей между ресурсами.

В [145] предлагается подход для автоматического выбора контрмер на основе показателей ущерба от атаки и эффективности реагирования. Для этого используется карта системы (граф зависимостей, объединяющий приоритетные ресурсы), иерархия ресурсов (группировка ресурсов по типам с выделением контрмер для каждого типа) и стоимостная модель (в которой ресурсам назначаются стоимости). Ущерб от атаки вычисляется как сумма стоимостей узлов системной карты, на которые атакующее действие повлияло негативно. Эффективность реагирования определяется как сумма стоимостей узлов, восстановленных в результате реализации контрмеры. И, наконец, побочный ущерб при реагировании определяется как стоимость узлов, на которые реализация контрмеры повлияла отрицательно. Далее необходимо выбрать набор контрмер, которые дадут наибольший выигрыш при наименьших затратах. Авторы исходят из того, что им известна полная картина вторжения, и оптимальная цепочка контрмер будет состоять из набора оптимальных контрмер для каждого узла (из альтернативных контрмер для каждого узла выбирается та, которая дает наибольший выигрыш). Кроме того, предлагается идея выбора оптимальных контрмер в условиях неопределенности (т.е. когда неизвестны точные действия злоумышленника) на основе теории принятия решений.

В [135] для определения эффективности реагирования применяется показатель жизнеспособности системы, зависящий от того, какое количество операций и целей безопасности системы может быть сохранено в случае вторжения.

При выборе узлов реализации защитных мер применяется индекс уверенности в компрометации (*CCI*), способ вычисления которого описан выше: если для узла $CCI>\tau$, где τ — пороговое значение, то данный узел является сильным кандидатом (SC) для реализации защитных мер; если $CCI\leq\tau$, но дальнейший обход по дугам типа И может достичь узла SC, то та-

кой узел является слабым кандидатом (WC); если $CCI \leq \tau$, но дальнейший обход по дугам любого типа может достичь узла SC, то такой узел является очень слабым кандидатом (VWC); в противном случае узел не является кандидатом.

Для выбранных узлов выбираются методы реагирования на основе показателя «индекс реагирования» (*Response Index*, *RI*): $RI=a \cdot EI - b \cdot DI$, где a и b — параметры развертывания мер, *EI* — индекс эффективности защитной меры, *DI* — индекс разрушений, наносимых защитной мерой системе. Выбирается защитная мера с максимальным *RI*.

В [147] получила дальнейшее развитие концепция производительности ресурсов, предложенная в [144]. Стоимость ущерба, нанесенного в результате проведения атаки, рассчитывается как значение доступности ресурсов после проведения атаки (в процентном отношении). При этом доступность ресурсов зависит от внутренней доступности ресурса и доступности зависимых ресурсов. Авторы предлагают расчет показателей для вторжения и реакции на вторжение на основе графа зависимостей между ресурсами. Авторы учитывают только доступность ресурсов $A(r)$, которая зависит от внутренней доступности ресурса и распространенной зависимости. Стоимость ущерба (Ущерб от Атаки) рассчитывается как значение доступности после проведения атаки (в процентном отношении). Кроме того, авторы рассчитывают такие показатели, как Затраты на Реакцию (нормализованное количество ресурсов, которые нужно изменить для реализации реакции), Эффективность Реакции (рассчитывается как разница между ущербом от атаки и от реакции) и Дополнительный Ущерб от Реакции (рассчитывается также, как ущерб от атаки, но на основе графа доступности, построенного после реализации реакции).

В [187] предлагается подход к определению затрат на реагирование на основе трех показателей: *OC* — затраты на ежедневную поддержку защитных мер; *RG* — качество реагирования, определяющее эффективность мер против обнаруженной атаки; *RSI* — влияние реагирования на функционирование системы. Затраты на реагирование определяются по формуле: $RC=OC+RSI-RG$.

Методология определения затрат на реагирование включает следующие шаги:

1. Классификация системы, которая позволит определить ее требования к безопасности.
2. Определение и ранжирование целей безопасности в терминах конфиденциальности, целостности и доступности на шкале от 0 до 1.
3. Выделение ресурсов (активы, сервисы и пользователи) и определение их весов в рамках целей безопасности:

$$W_{SR} = \sum_i [SRimportance_i \times PolicyCategoryWeight_i],$$

где i — цель безопасности,

$SRimportance_i$ — важность ресурса для i-й цели безопасности,

$PolicyCategoryWeight_i$ — вес данной цели для системы; определение набора средств реагирования для системы (мер, которые могут вернуть ресурс в защищенное состояние, в случае различных атак).

4. Определение затрат на ежедневную поддержку защитных мер (человеческие ресурсы, системные ресурсы, прямые затраты) на основе экспертных знаний.

5. Определение качества реагирования на основе выделения возможных мер реагирования и определения показателя эффективности каждой меры против известных атак по формуле:

$$RG_{R_i}(I_j) = \sum_{k\in[1\ldots n]} Avail(SR_k^j) \times W_{SR_k},$$

где R_i — мера реагирования, $i \in [1 \ldots m]$, m — количество различных мер реагирования,

I_j — атака,

SR_k^j — ресурс, затронутый при атаке,

n — количество затронутых ресурсов,

$Avail(SR_k^j)$ — доступность ресурса SR_k^j для атаки.

6. Вычисление ущерба системы при реагировании основано на выделении ресурсов, которым наносится ущерб при реагировании, ранжировании мер реагирования для ресурса в зависимости от уровня ущерба, вычислении ущерба, наносимого ресурсу при реагировании по формуле:

$$RSI_{R_i} = \sum_{SR} Impact_{R,SR} \times W_{SR},$$

где $Impact_{R,SR} = 1 - \frac{i}{m}$, i — порядковый номер меры реагирования, m — общее количество мер реагирования.

Недостатком данного подхода является то, что он требует больших временных затрат, так как большую часть действий необходимо выполнять вручную.

В [148] предлагается показатель выбора контрмер для реагирования на атаки на основе графов зависимостей сервисов — *Показатель возврата инвестиций в реагирование* (*Return-On-Response-Investment*, *RORI*):

$$RORI = \frac{RG-(CD+OC)}{CD+OC},$$

где RG — эффективность реагирования, $RG = IC_b - IC_a$,

IC_b — ожидаемые негативные последствия атаки, в случае отсутствия контрмер;

IC_a — ожидаемые негативные последствия атаки, в случае реализации контрмер;

CD — побочные потери при реагировании;

OC — затраты на контрмеры.

Для реализации выбирается контрмера с наибольшим значением показателя *RORI*.

Так как на практике сложно вычислить отдельно побочные потери при реагировании и ожидаемые негативные последствия атаки, в случае реализации контрмер, формула для вычисления показателя *RORI* преобразуется следующим образом:

$$RORI = \frac{(IC_b-IC_a)-(CD+OC)}{CD+OC} = \frac{IC_b-(IC_a+CD)-OC)}{CD+OC} = \frac{(IC_b-RC)-OC)}{CD+OC},$$

где RC — совместные негативные последствия кибератаки и реализации контрмеры.

Преимуществом данного подхода является учет распространения ущерба через зависимости сервисов. Недостатком подхода является то, что не учитывается неопределенность успешности атакующих действий.

В [189] рассмотрены недостатки показателя *RORI* (не учитывается вариант отсутствия контрмер и размер инфраструктуры системы), и предложен улучшенный показатель *RORI*:

$$RORI = \frac{(ALE \times RM) - ARC}{ARC + AIV} \times 100,$$

где *ALE* — ожидаемые годовые потери (соответствует последствиям негативного события в случае отсутствия контрмер), которые зависят от критичности и правдоподобности атаки;

RM — уровень снижения риска в случае реализации контрмеры;

ARC — ожидаемые годовые затраты на новую контрмеру, *ARC=CD+OC*;

AIV — годовые затраты на инфраструктуру (оборудование, поддержка), в случае реализации защитной меры.

В [191] контрмеры оцениваются по трем параметрам (выигрышу от реализации контрмеры, затратам на реализацию контрмеры и дополнительной пользе от реализации контрмеры), на основе которых определяется общий выигрыш от реализации контрмеры.

Общий выигрыш от реализации *k*-й контрмеры:

$Net\_Benefit_k = Benefit_k - Added\_Cost_k + Added\_Profit_k, \forall k = \{1,2,3, \dots, l\}$,

где l — количество контрмер,

$Benefit_k$ — выигрыш от реализации *k*-й контрмеры, $Benefit_k = ALE_0 - ALE_k$, $\forall k = \{1,2,3, \dots, l\}$, k=0 — статус-кво,

ALE — произведение частоты возникновения инцидента (в год) на потери,

$\sum_{i=1}^{n}\{F_0(B_i) \times D_0(B_i) \times \prod_{j=1}^{m}[(1 - E_f(B_i, S_j) \times I_k(S_j)) \times (1 - E_d(B_i, S_j) \times I_k(S_j))]\}$,

n — количество инцидентов, $i = \{1,2,3,..., n\}$,

m — количество контрмер, $i = \{1,2,3,..., m\}$,

$F_0(B_i)$ — исходная оценка частоты возникновения инцидента B_i,

$D_0(B_i)$ — исходная оценка последствий инцидента B_i,

$E_f(B_i, S_j)$ — снижение частоты возникновения инцидента B_i в результате реализации контрмеры S_j,

$I_k(S_j)$ — бинарная функция, определяющая, входит ли контрмера S_j в политику P_k,

$E_d(B_i, S_j)$ — снижение последствий инцидента B_i в результате реализации контрмеры S_j,

$Added\_Cost_k$ — затраты на *k*-ю контрмеру, $Added\_Cost_k = \sum_{j=1}^{m} C(S_j) \times I_k(S_j)$,

$C(S_j)$ — затраты на контрмеру S_j,

$Added\_Profit_k$ — дополнительная польза от реализации *k*-й контрмеры, $Added\_Profit_k = \sum_{j=1}^{m} R(S_j) \times I_k(S_j)$,

$R(S_j)$ — доходы от реализации контрмеры S_j.

В [185] рассматривается подход к минимизации затрат на повышение уровня защищенности сетей на основе графов зависимостей между экс-

плоитами (exploit dependency graphs). Авторы преобразуют путь на графе до цели атаки в выражение, включающее начальные предусловия атаки в конъюнктивной нормальной форме. Далее для реализации защитных мер выбираются дизъюнкции, включающие максимальное количество переменных, соответствующих начальным предусловиям атаки. Стоимость учитывается для выбора набора защитных мер с минимальной стоимостью.

Недостатком является то, что при выборе защитных мер учитываются только начальные узлы.

4.2. Программные средства выбора контрмер

Система Cauldron. Система Cauldron от компании CyVision [193] является средством визуализации и моделирования угроз, предназначенным для проактивного управления защитой в глубину КС. Cauldron использует технологию нейросетей для анализа КС любого масштаба. Инструмент позволяет учитывать программно-аппаратное обеспечение КС, конфигурации и правила контроля доступа, а также позволяет обеспечить соответствие стандартам HIPAA, PCI, ISO 27001 или другим.

Для выбора защитных мер, позволяющих минимизировать риски и защитить ценные активы, используется подход, основанный на CVSS, а также анализ путей угроз с учетом связей между хостами и различных конфигураций. На основе данных подходов Cauldron позволяет выявить и оценить уязвимости и ранжировать контрмеры с точки зрения эффективности по стоимости. Для этого используется показатель ROI.

Для оценки используются различные показатели, разделенные на группы [194]: *Преследования* (victimization) — измеряет риск с точки зрения уязвимостей системы; *Размерные* — определяют размер графа; *Сдерживающие* — измеряют риск с точки зрения минимизации негативного воздействия уязвимости; *Топологические* — измеряют риск с точки зрения свойств графа (связность, циклы, глубина).

Категория *Преследования* включает показатели: *Existence* (существование) — относительное количество уязвимых сетевых сервисов, на шкале (0,10); *Exploitability* (эксплуатируемость) — среднее значение показателя CVSS *Exploitability* среди всех уязвимостей хостов, на шкале (0,10); *Impact* (ущерб) — среднее значение показателя CVSS *Impact* (относительный ущерб эксплуатации) для всех уязвимостей сети, на шкале (0,10).

Показатель *Existence* вычисляется по формуле: $Existence=10s_v/(s_v+s_n)$, где s_v — уязвимые сервисы, s_n — неуязвимые сервисы. Показатель *Эксплуатируемость* всей сети вычисляется по формуле:

$$Exploitability = \sum_i^{|U|} CVSS_{Exploitability}(u_i)/|U|,$$

где $CVSS_{Exploitability}(u_i)$ — CVSS *Exploitability* уязвимости u_i;

$|U|$ — количество уязвимостей в сети.

Показатель *Impact* вычисляется по формуле:

$$Impact = \sum_i^{|U|} CVSS_{Impact}(u_i)/|U|,$$

где $CVSS_{Impact}(u_i)$ — CVSS *Impact* для уязвимости u_i;

$|U|$ — количество уязвимостей в сети.

Категория *Размерные* введена, так как авторы предполагают прямую связь между размером графа и риском, поскольку чем больше граф, тем больше существует путей компрометации сети. Данная категория включает показатели: *Attack Vector*s (вектора атаки) — количество векторов атаки из одного шага относительно общего возможного количества для сети, на шкале (0,10).

Показатель *Attack Vector*s вычисляется по формуле:

$$AttackVectors = 10\sqrt{v_a, v_p},$$

где v_a — вектора атаки,

v_p — возможные векторы атаки (исходя из количества открытых портов хостов сети) [194].

Система NETSPA. Система NETSPA [165, 182] реализована на C++ и включает следующие компоненты:

- база данных действий, программного обеспечения и конфигураций сети;
- модели объектов, которые формируются на основе информации из баз данных;
- вычислительный механизм, формирующий граф атак и реализующий методики оценивания защищенности и выбора контрмер, описанные в предыдущих разделах;
- подсистема визуализации.

Программное обеспечение SecurITree. ПО SecurITree от компании Amenaza [184] использует деревья атак для анализа рисков и выбора контрмер. Выбор контрмер осуществляется путем формирования дерева атак до и после внедрения контрмер и сравнения полученных оценок риска — если контрмера позволяет снизить риски ИБ, она рекомендуется для внедрения.

Выводы по главе 4

В области выбора контрмер для проактивного и реактивного реагирования на кибератаки существует большое количество методик, основанных на теории игр, на логическом выводе на графах атак, на байесовских графах атак и др. К преимуществам методик, основанных на графах, относится возможность прогнозирования кибератак для своевременного принятия решений по выбору контрмер, а также возможность анализа и учета предыдущих шагов атаки. При объединении с экономическими индексами такие методики позволяют связать принятие решений с финансовыми затратами и выигрышем для организации. Однако результаты выполнения таких методик сильно зависят от результатов анализа защищенности. А также необходимо повышать оперативность методик принятия решений в реальном времени.

Глава 5. Методики и средства оценивания защищенности и выбора контрмер для управления кибербезопасностью, основанные на графах атак и зависимостей сервисов

5.1. Показатели защищенности

Авторами была разработана система оценивания защищенности и выбора контрмер для управления кибербезопасностью в составе SIEM-систем, учитывающая достоинства и недостатки показателей и методик, рассмотренных в предыдущих главах [195]. В основе разработанной системы лежит классификация и комплекс показателей защищенности, используемых для оценивания защищенности компьютерных систем и сетей, выбора контрмер и мониторинга защищенности [195]. Предложенная авторами классификация показателей защищенности разработана с учетом следующих требований:

- классификация должна соответствовать последним исследованиям в области показателей защищенности;
- классификация должна учитывать различные объекты оценивания защищенности;
- показатели должны рассчитываться на основе подхода к анализу защищенности, применяющего в качестве входных данных графы атак и графы зависимостей сервисов;
- показатели должны позволять получить оценку текущей ситуации по защищенности в любой момент времени на основе доступных входных данных с разным уровнем детализации;
- показатели должны учитывать информацию, поступающую от SIEM-систем;
- показатели должны позволять выбрать контрмеры в любой момент времени на основе доступных входных данных;
- классификация должна позволять выделить статический и динамический режимы оценивания защищенности и выбора контрмер;
- показатели должны позволять оценить стоимостные характеристики атак и контрмер.

Были выделены следующие уровни классификации: уровень инфраструктуры; уровень графа атак; уровень атакующего; уровень событий; уровень выбора контрмер и уровень системы (интегральный уровень). Каждая категория включает следующие подкатегории: базовые характеристики, стоимостные характеристики, характеристики нулевого дня. Внутри каждой подкатегории выделяются основные показатели, используемые для вычисления значения уровня риска, позволяющего определить уровень защищенности компьютерной системы или сети от кибератак, и вспомогательные, не используемые для вычисления значения уровня риска.

Для удовлетворения требований адекватного отражения текущей ситуации на основе доступных входных данных и применения в качестве входных

данных графов атак и графов зависимостей сервисов, уровни классификации выделены в зависимости от обрабатываемых входных данных: знаний об анализируемой компьютерной системе или сети, ее уязвимостях, атакующем, или происходящих в системе событиях безопасности. Для удовлетворения требованиям интеграции с SIEM-системами и выделения статического и динамического режимов оценивания защищенности и выбора контрмер введен уровень событий, принимающий в качестве входных данных информацию о событиях безопасности. Для учета контрмер при оценивании защищенности, а также выбора контрмер, добавлен уровень выбора контрмер. Для учета стоимостных характеристик атак и контрмер введена категория стоимостных характеристик. Каждый уровень содержит набор показателей, отражающий последние исследования в данной области.

На рисунке 5.1 представлен предлагаемый комплекс показателей защищенности с учетом разработанной классификации. Для вычисления интегральных показателей достаточно показателей уровня инфраструктуры. В дальнейшем значения интегральных показателей можно уточнять показателями остальных уровней, что показано пунктирными стрелками.

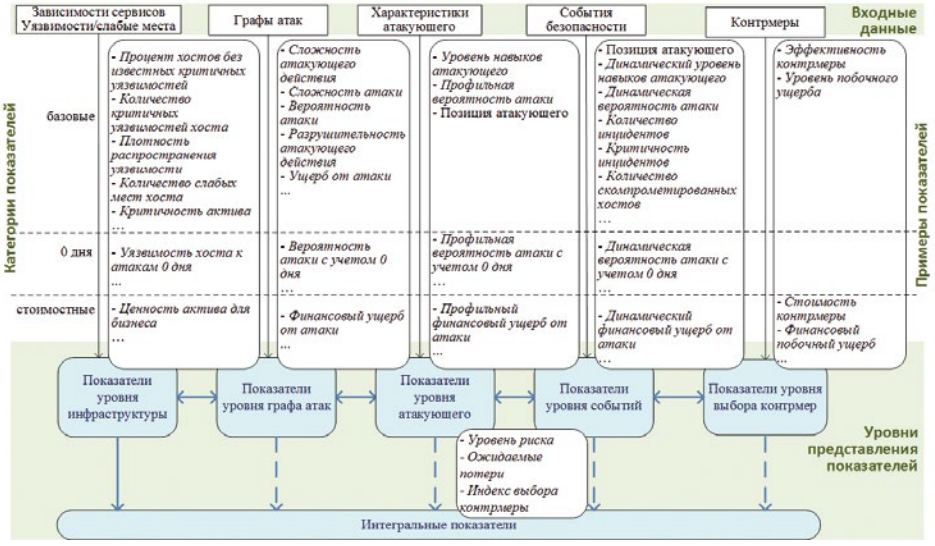


Рис. 5.1. Комплекс показателей защищенности

В литературе даются различные определения показателей защищенности. Под показателем защищенности будем понимать систему взаимосвязанных измерений, позволяющих квантифицировать отдельную характеристику, то есть, это измерение, которое сравнивается со шкалой или критерием для получения значимого результата [197, 198]. Показатели защищенности необходимы для отслеживания ситуации по защищенности и облегчения выполнения задач по улучшению ситуации путем применения соответствующих действий [199]. Показатели являются результатом анализа (в отличие от измерений), и для выполнения поставленных целей они должны удовлетворять следующим требованиям [197]: показатели должны быть ценными

для целевой аудитории; стоимость измерений не должна превышать ценность показателей; своевременность и частота измерений должна соответствовать частоте изменений объекта измерений; показатели должны быть объективными и количественно выраженными; показатели должны быть воспроизводимыми разными экспертами в одинаковых условиях.

На первом уровне, *инфраструктуры*, для вычисления показателей доступны входные данные: модель компьютерной системы [166]; информация о хостах; характеристики программно-аппаратного обеспечения, в том числе уязвимости; характеристики уязвимостей на основе открытой базы уязвимостей NVD [13]; характеристики слабых мест хоста на основе базы слабых мест CWE [107]; сервисы компьютерной системы; модель зависимостей сервисов.

Исходя из особенностей входных данных, используемых в исследовании, и на основе показателей из [131], были выбраны следующие вспомогательные базовые показатели: *процент хостов без известных критичных уязвимостей* (*Percent of Hosts with No Known Severe Vulnerabilities*, *PHNKSV*); *количество критичных уязвимостей хоста* (*Number of Known Severe Host Vulnerability Instances*, *NKSHVI*); *нормированное количество критичных уязвимостей хоста* (*Normalized Number of Known Severe Host Vulnerability Instances*, *NNKSHVI*), *плотность распространения уязвимости* (*Target Vulnerability Distribution*, *TVD*). И добавлены показатели: *количество слабых мест хоста* (*Number of Known Host Weakness Instances*, *NKHWI*); *нормированное количество слабых мест хоста* (*Normalized Number of Known Host Weakness Instances*, *NNKHWI*). Вспомогательные показатели 0 дня: *уязвимость хоста к атакам 0-го дня* (*Host Vulnerability to Zero-Day Attacks*).

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

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

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

Были выбраны следующие основные базовые показатели: *сложность атакующего действия* (*Complexity of Attack Action*); *сложность атаки* (*Attack Complexity*); *вероятность атаки* (*Attack Potentiality*); *разрушительность атакующего действия*; *ущерб от атаки* (*Attack Impact*). Основные стоимостные показатели: *финансовый ущерб от атаки* (*Monetary Attack*

Impact). Основные показатели 0 дня: *вероятность атаки с учетом уязвимостей 0 дня* (*Attack Potentiality Considering Zero-Days*).

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

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

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

Были выбраны следующие вспомогательные базовые показатели: *количество инцидентов* (*Number of Incidents*). Данный показатель основан на показателе CIS *количество инцидентов* [131] и модифицирован для реализуемой системы следующим образом: *количество инцидентов на хосте* (*Number of Host Incidents*); *нормированное количество инцидентов на хосте* (*Normalized Number of Host Incidents*); *количество инцидентов в системе* (*Number of System Incidents*). И добавлены показатели: *средняя критичность инцидентов на хосте* (*Mean Criticality of Host Incidents*) и *средняя критичность инцидентов в системе* (*Mean Criticality of System Incidents*); *количество скомпрометированных хостов* (*Compromised Hosts*).

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

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

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

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

Были выбраны следующие показатели: *уровень риска атаки* (*Attack Risk Level*); *уровень риска сервиса* (*Service Risk Level*); *уровень риска хоста* (*Host Risk Level*); *уровень риска КС* (*System Risk Level*); *индекс выбора контрмеры*

(*Countermeasure Selection Index*). Стоимостные показатели: *ожидаемые потери* (*Loss Expectancy*).

5.2. Методика оценивания защищенности на основе графов атак и зависимостей сервисов

Авторами разработана методика оценивания защищенности на основе комплекса показателей защищенности [200–202]. Схема подхода, лежащего в основе методик оценивания защищенности и выбора контрмер представлена на рисунке 5.2. Подход включает три этапа: (1) сбор входных данных; (2) вычисление показателей защищенности; (3) определение уровня защищенности. Полученные на каждом этапе выходные данные используются как входные данные следующего этапа. Перемещение данных между этапами показано стрелками. Верхний уровень А рисунка 5.2 соответствует входным данным, применяемым для вычисления показателей.

Жирным курсивом выделены названия стандартов, применяемых для представления входных данных. Стандарты представления входных данных, входящие в протокол SCAP (подробно описанный в главе 2), применяются для удовлетворения требования автоматизации оценивания защищенности и выбора контрмер. Стандарт CVE используется для представления уязвимостей, CPE — для представления ПО, CCE — для представления конфигураций, CRE и ERI — для представления контрмер. Стандарт CVSS используется для оценивания уязвимостей. На основе собранных данных строятся модели, которые потом применяются для вычисления показателей (уровень В). Выходные данные отображены на уровне Е. На уровнях С и D представлены данные, применяемые на соответствующих этапах подхода. Данные разделены на группы в зависимости от источников, в соответствии с уровнями системы показателей. Это позволяет выделить статический и динамический режимы работы методики. Статическому режиму работы соответствуют уровень инфраструктуры, уровень графа атак и атакующего.

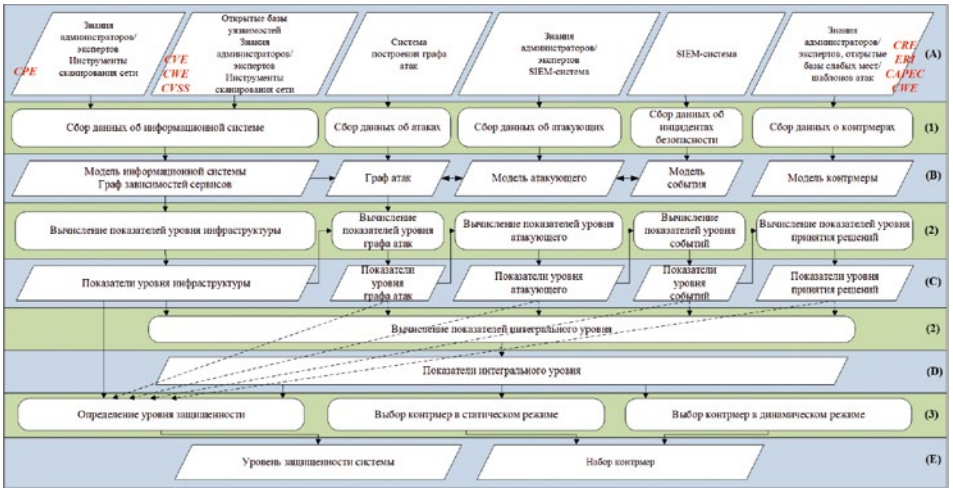


Рис. 5.2. Обобщенная схема подхода к оцениванию защищенности и выбору контрмер

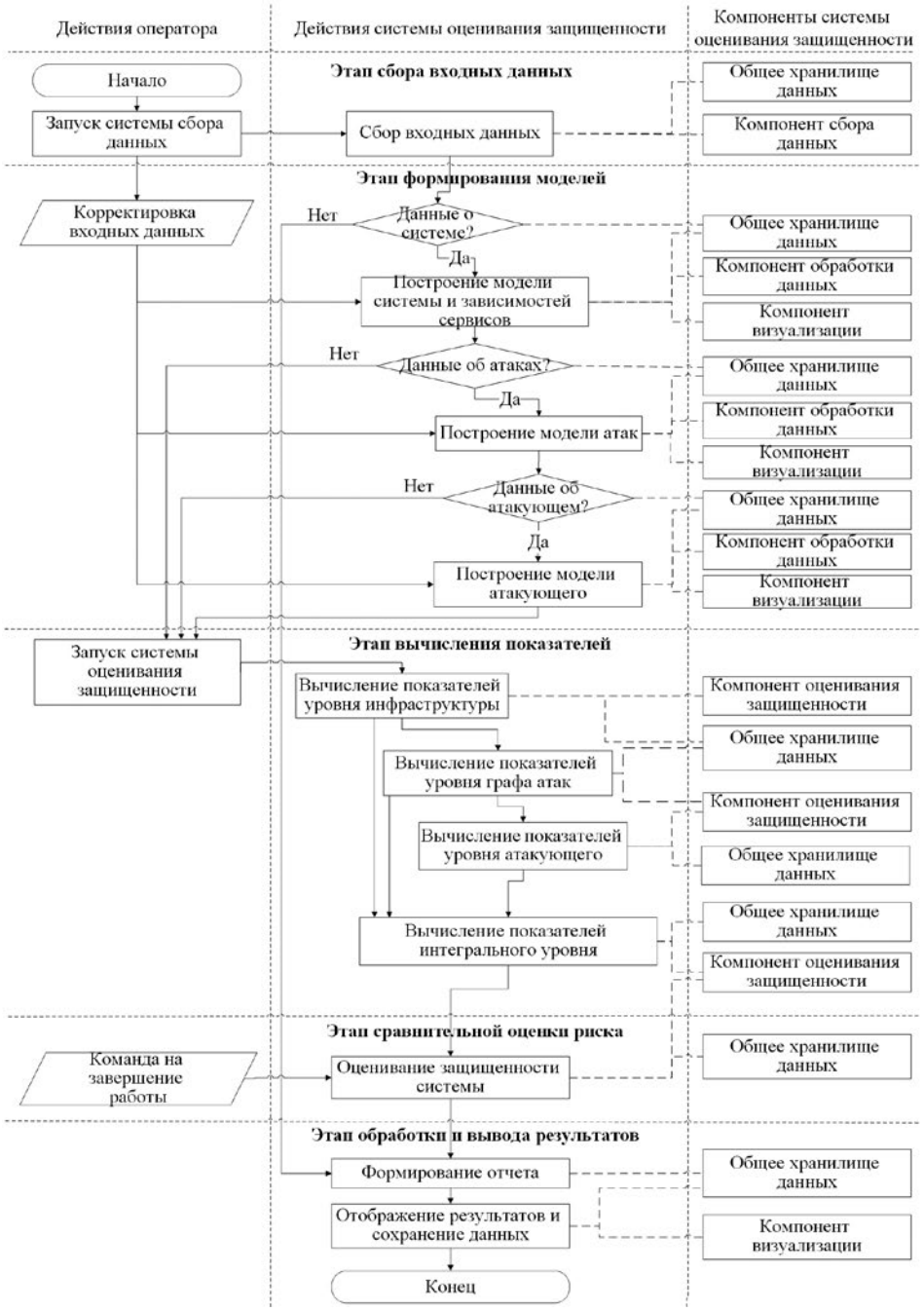


Рис. 5.3. Схема методики оценивания защищенности в статическом режиме

Более детально этапы работы методики в статическом режиме и последовательность их выполнения представлены на рисунке 5.3. Результат работы методики в статическом режиме: уровень риска компьютерной си-

стемы и комплекс показателей защищенности. Для динамического режима работы дополнительно введен уровень событий. Более детально этапы работы методики в динамическом режиме представлены на рисунке 5.4. Результат работы в динамическом режиме: путь атаки, цель атаки, характеристики атакующего, ожидаемые потери в случае успешной реализации атаки (уровень риска).

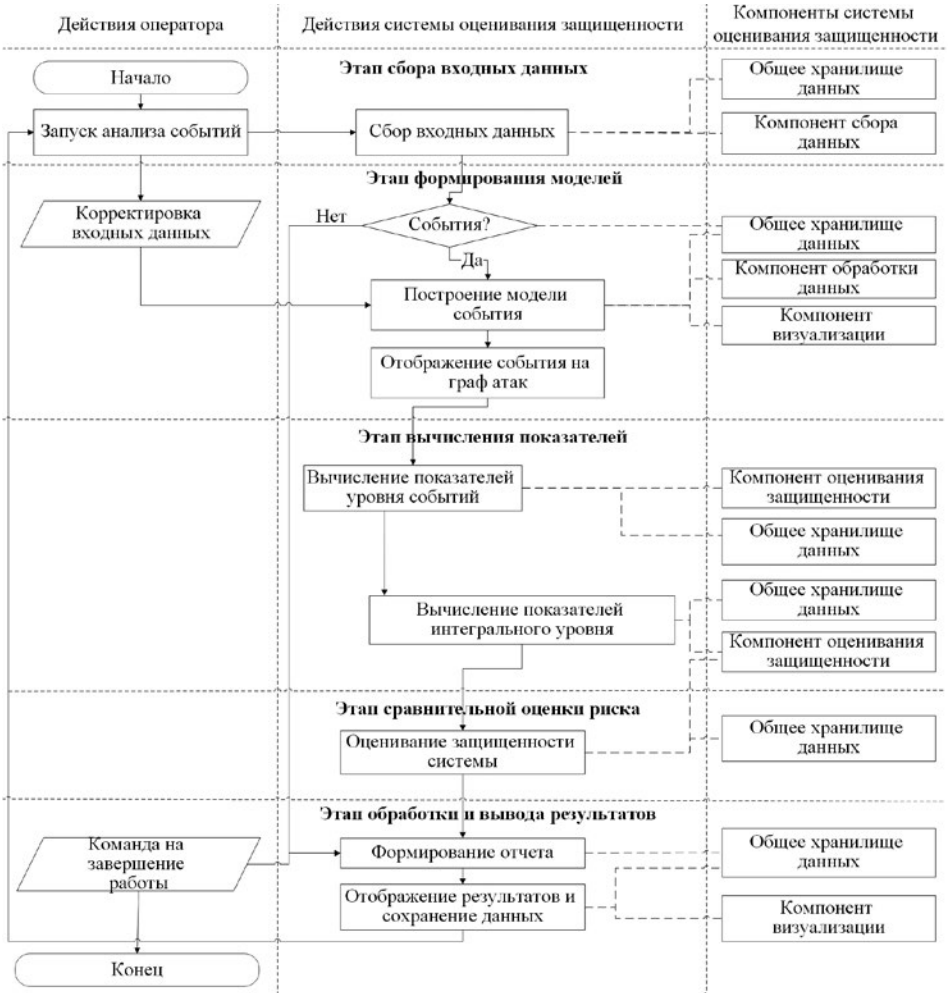


Рис. 5.4. Схема методики оценивания защищенности в динамическом режиме

Применяемый подход относится к так называемым «any-time» подходам. Основной особенностью подхода является возможность получения оценок и выбора контрмер на каждом уровне системы показателей. При получении новых данных (других уровней) оценки могут быть улучшены применением алгоритмов соответствующего уровня. Общие показатели оценки защищенности и выбора контрмер являются показателями интегрального уровня. Для их вычисления необходимы показатели хотя бы топологическо-

го уровня (все остальные уровни являются дополнительными, что показано на рисунке 5.2 пунктирными стрелками).

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

Вычисление вспомогательных показателей уровня инфраструктуры. Для вычисления вспомогательных показателей уровня инфраструктуры применяются данные о хостах компьютерной системы, уязвимостях их программно-аппаратного обеспечения и характеристики уязвимостей, полученные из базы NVD [13].

Показатель *процент хостов без известных критичных уязвимостей PHNKSV* основан на показателе CIS *процент систем без известных критичных уязвимостей* (*Percent of Systems with No Known Severe Vulnerabilities*) [131], который был переопределен для хостов. Данный показатель отражает относительную уязвимость системы. Критичными в данном случае называются уязвимости со значением базовой оценки CVSS «Высокая». Показатель измеряется в процентах на шкале от 0 до 100 и может изменяться во времени. Формула для вычисления показателя:

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

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

n — общее количество хостов в сети.

Показатели *количество критичных уязвимостей хоста NKSHVI* и *нормированное количество критичных уязвимостей хоста NNKSHVI* основаны на показателе CIS *количество известных уязвимостей* (*Number of Known Vulnerability Instances*) [131]. Критичными в данном случае называются уязвимости со значением базовой оценки CVSS «Высокая». Показатель *количество критичных уязвимостей хоста* может изменяться во времени.

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

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

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

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

Показатель *количество слабых мест хоста NKHWI* может изменяться во времени. Данный показатель рассчитывается на основе слабых мест программно-аппаратного обеспечения хоста в соответствии со словарем CWE [106]. Показатель *нормированное количество слабых мест хоста NNKHWI* отображает количество слабых мест хоста по отношению к другим хостам и может изменяться во времени:

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

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

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

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

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

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

n — общее количество хостов в сети.

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

Алгоритмы вычисления основных показателей уровня инфраструктуры. Для вычисления уровня риска используется определение, данное в стандарте [36], согласно которому риск характеризуется комбинацией вероятности возникновения инцидента (или проведения атаки) и его разрушительного воздействия. Последствия в случае успешной реализации атаки (разрушительное воздействие инцидента) зависят от ценности (критичности) актива для его владельцев и разрушительности атакующего действия. Успешность атаки (вероятность возникновения инцидента) зависит от наличия уязвимости, которая позволит осуществить атаку, наличия доступа к этой уязвимости, сложности ее эксплуатации, возможностей атакующего и привлекательности уязвимости (то есть ее важности для достижения цели атакующего). Поэтому в качестве основных были выбраны следующие показатели: *критичность актива*; *разрушительность атакующего действия*; *ущерб от атакующего действия*; *вероятность атакующего действия*. Стоимостные показатели, используемые для вычисления значения уровня риска: *ценность актива для бизнеса* (*Business Value of the Asset*); *финансовый ущерб* (*Monetary Impact*).

Алгоритм оценивания критичности активов ИС. Критичность актива определяет важность конкретного актива для целей и миссии организации. Критичность актива измеряется по шкале от 0 до 100. Для оценки критичности активов предлагается объединить операции организационного уровня, описанные в стандарте [39], и технического уровня, предлагаемые в ряде исследовательских работ [136, 144, 148]. То есть перейти от целей и миссии организации к информационным активам, которые их поддерживают, и связанным с ними угрозам и уязвимостям. Для этого выделено два этапа алгоритма: этап организационного уровня и этап технического уровня (рисунок 5.5).

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

Входные данные первого этапа работы алгоритма: идентифицированные ИТ-активы (определяются владельцами активов), критерии оценки активов, шкала оценок.

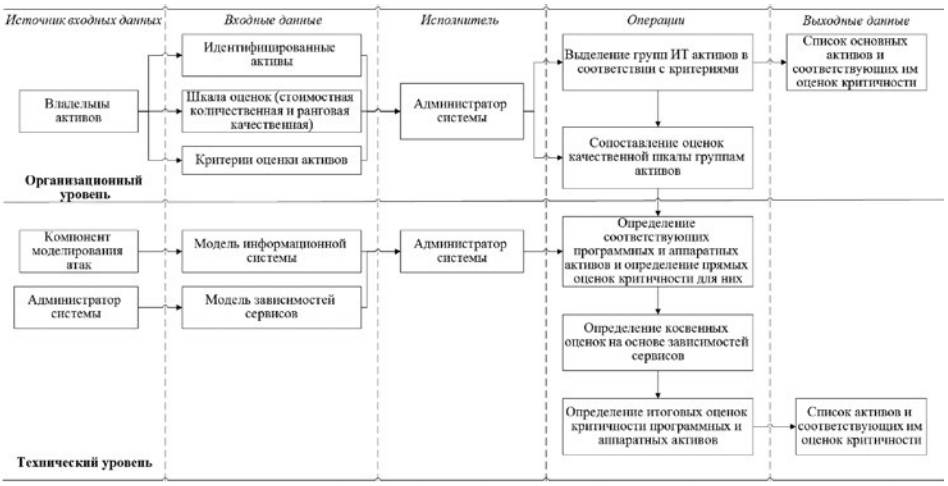


Рис. 5.5. Обобщенная схема алгоритма оценивания критичности активов

Согласно [39] определение ценности активов осуществляется на основе восстановительной стоимости актива и последствий для бизнеса от потери или компрометации актива. Критерии последствий для бизнеса от потери или компрометации актива (нарушение конфиденциальности, целостности и доступности) должны разрабатываться с учетом уровня классификации актива, нарушения ИБ, нарушения оперативной деятельности, потери ценности бизнеса и финансовой ценности, нарушения планов и конечных сроков, ущерба для репутации, нарушения требований. Критерии оценивания активов могут определяться, например, на основе [39]. В соответствие критериям ставится количественная шкала оценок, отражающая финансовые потери в случае потери конфиденциальности, целостности и доступности активов (то есть прямые потери в случае нарушения критерия и последующие затраты на восстановление). Для этого может использоваться шкала, предложенная в [203] для правительственных организаций и одобренная для организаций, использующих КС, в рамках проекта MASSIF [87] (таблица 5.1). Например, реализация угрозы нарушения конфиденциальности личных данных (активом в данном случае выступают данные), ведет к негативному воздействию на репутацию компании и может потребовать значительных затрат на ее восстановление, что соответствует уровню критичности «Значительная» (€10,000) в таблице 5.1.

В рамках разработанного алгоритма количественной шкале ставится в соответствие качественная, на основе которой определяются оценки активов от 0 до 100 (чтобы сохранить отношение между различными уровнями критичности). Для данной шкалы потери в случае реализации угрозы для каждого следующего уровня сравнимы с потерями, соответствующими десяти случаям реализации угрозы предыдущего уровня (что необходимо

учитывать при формировании шкалы). Стоимостная количественная шкала может отличаться для разных организаций (сохраняя диапазон от нулевых затрат до годового бюджета организации), однако качественная шкала остается неизменной.

Таблица 5.1

Шкалы оценки критичности организационных активов

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

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

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

На втором этапе, технического уровня, определяются программно-аппаратные активы системы, необходимые для поддержания основных активов организации и прямые и внешние оценки их критичности. Входные данные второго этапа работы алгоритма: модель компьютерной системы или сети, модель зависимостей сервисов и результаты работы первого этапа. В качестве модели компьютерной системы или сети может использоваться, напри-

мер, модель сети, предложенная в [166], которая включает список моделей хостов (определяемых списком программного и аппаратного обеспечения и политиками), список связей между хостами и тип зависимости хостов. Данная модель расширена путем добавления в модель хостов списка сервисов. Под сервисом будем понимать ресурс, предоставляющий возможность выполнения задач, формирующих необходимую функциональность с точки зрения поставщиков и потребителей услуг [171]. Модель сервиса определена следующим образом: $R = (T,Cr)$, где T — тип сервиса (*ИТ-активы, порт или программно-аппаратное обеспечение*). Cr — критичность сервиса; $Cr = [Cr_r(c)\ Cr_r(i)\ Cr_r(a)]$, где $Cr_r(c)$, $Cr_r(i)$, $Cr_r(a)$ — критичность сохранения свойств конфиденциальности, целостности и доступности сервиса r, соответственно.

Модель зависимостей сервисов задается следующим образом [204]: $SG = (R, L, \varepsilon)$, где R — множество узлов графа зависимостей сервисов (сервисов), L — множество связей ($L \subseteq R \times R$), ε — множество кортежей, определяющих тип зависимости между сервисами, вида $<L_k, d_k>$, где $L_k \in L$, $d_k \in \{\text{И, ИЛИ}\}$. Связь определяется как $L = (r_i, r_j, W)$, где $r_i, r_j \in R$, $r_j \in Det(r_i)$, $Det(r_i)$ — множество всех прямых потомков сервиса r_i (то есть сервисов, от свойств безопасности которых напрямую зависят свойства безопасности r_i), W — весовая матрица, определяющая степень зависимости свойств безопасности сервиса-предка от свойств безопасности сервиса-потомка. В [146] авторы выделяют структурные зависимости (между сервисами различных уровней модели ISO/OSI) и функциональные (между разными сервисами одного уровня). Пример структурных зависимостей: зависимость между веб-приложением, сервером JBoss и портом tcp/443 на рисунке 5.6. Пример функциональных зависимостей: зависимость между веб-приложением и аутентификацией на рисунке 5.6.

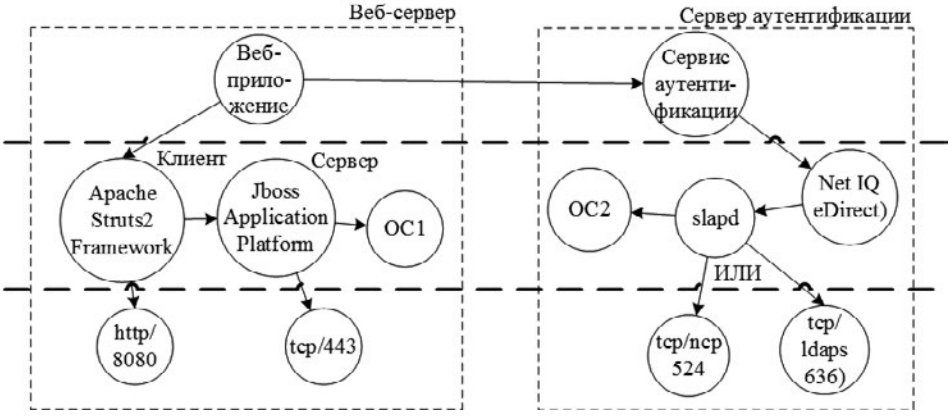


Рис. 5.6. Пример функциональных и структурных зависимостей [146]

Данная модель была выбрана, так как в отличие от других работ на эту тему, в ней учитывается не только распространение ущерба доступности, но и распространение ущерба целостности и конфиденциальности. В [146] модель зависимостей сервисов применяется для определения ущерба, рас-

пространяемого через зависимости сервисов, в динамическом режиме работы системы. В рамках разработанных методик модель зависимостей сервисов предлагается использовать для определения показателя критичности активов в статическом режиме работы. Это связано с тем, что для определения распространения атаки в системе используется граф атак, и необходимо связать распространение ущерба через зависимости сервисов с узлами графа атак (это позволит учитывать как распространение атаки в системе, так и распространение ущерба). Кроме того, это позволит экономить время в динамическом режиме работы.

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

Прямые оценки критичности сервисов системы по параметрам конфиденциальности, целостности и доступности $\langle I\_Criticality_{r_k}(c), I\_Criticality_{r_k}(i), I\_Criticality_{r_k}(a)\rangle$ назначаются вручную экспертами. Для определения внешних оценок критичности $\langle P\_Criticality_{r_k}(c), P\_Criticality_{r_k}(i), P\_Criticality_{r_k}(a)\rangle$ на основе зависимостей сервисов необходимо сформировать модель зависимостей сервисов. Сервисы частично определяются при помощи средств сканирования сети: активных (таких, как Nmap [205] и Nessus [206]) и пассивных (таких, как Wireshark [207]), и частично вручную администраторами. Программно-аппаратные сервисы задаются ссылками на соответствующее программно-аппаратное обеспечение (определенное с использованием словаря CPE). Зависимости между сервисами задаются вручную администраторами.

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

Внешняя критичность сервиса зависит от прямой критичности сервиса r_k $\langle I\_Criticality_{r_k}(c), I\_Criticality_{r_k}(i), I\_Criticality_{r_k}(a)\rangle$ и критичности зависимых сервисов (сервисов-предков) $r_m \in Ant(r_k)$; $Ant(r_k)$ — множество всех прямых предков сервиса r_k (то есть сервисов, напрямую зависимых от свойств безопасности r_k). Внешняя критичность сервиса определяется следующим образом. Выделяются поддеревья всех ИТ-ресурсов (то есть корней графа зависимостей сервисов, с которых начинается работа алгоритма), так как критичность каждого сервиса системы зависит именно от них (они определяют финансовые потери организации в случае нарушений

защищенности). Поддерево задается дополнительным индексом в модели сервиса. Внутри одного поддерева (то есть потомки одного ИТ-ресурса) критичность любого сервиса-потомка по трем параметрам (конфиденциальности, целостности и доступности) не может превысить критичности корневого сервиса. Для корневого сервиса r_0 внешняя критичность равна его прямой критичности:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

где m — сервисы-предки из разных поддеревьев.

Если у сервиса-предка несколько сервисов-потомков, то распространение критичности зависит от типа зависимости: для конъюнктивных зависимостей критичность распространяется линейно в соответствии с формулой выше; для дизъюнктивных зависимостей критичность делится между сервисами-потомками:

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

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

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

где K — количество дизъюнктивных связей.

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

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

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

Алгоритм оценивания ущерба от атакующего действия. Ущерб от атакующего действия определяется на основе критичности актива и разрушительности атакующего действия. Разрушительность атакующего действия определяется на основе базовых показателей CVSS для уязвимостей [97]: влияние на конфиденциальность, целостность и доступность. Данный показатель определяет влияние на свойства безопасности актива R_k($k \in [1,l]$, l — количество всех программных активов организации) атакующего действия a_i в результате успешной эксплуатации уязвимости v_i ($i \in [1,m]$, m — множество всех уязвимостей данного актива), в виде трехзначного вектора $[ConfImpact_{k,i}(c) \quad IntegImpact_{k,i}(i) \quad AvailImpact_{k,i}(a)]$, где $ConfImpact_{k,i}(c)$ — влияние на конфиденциальность актива R_k,

$IntegImpact_{k,i}(c)$ — влияние на целостность актива, $AvailImpact_{k,i}(c)$ — влияние на доступность актива. $ConfImpact_{k,i}(c)$, $IntegImpact_{k,i}(c)$ и $AvailImpact_{k,i}(c)$ могут принимать значения {0; 0,275; 0,66} в соответствии с возможными значениями показателей CVSS *влияние на конфиденциальность*, *влияние на целостность* и *влияние на доступность*.

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

Алгоритм оценивания вероятности атакующего действия. Для определения *вероятности атакующего действия* a_i, использующего уязвимость v_i, используется показатель CVSS *Exploitability*:

$$Exploitability = 20 \times AccessVector \times AccessComplexity \times Authentication,$$

где *AccessVector — вектор доступа*; *AccessComplexity — сложность доступа*; *Authentication — аутентификация*.

Алгоритмы вычисления показателей уровня графа атак. Модель атак, используемая для вычисления показателей уровня графа атак [202, 208], является развитием моделей, предложенных в [166, 209]. Она задается следующим образом: *G*=(*S*, *L*, *Pc*), где *S* — множество узлов графа (атакующих действий), *L* — множество связей (*L*Нгруппа 1: *AV*=N/A (N — сетевой доступ, A — доступ из смежной сети), *priv*=user/other (привилегии пользователя или другие), *CIA*=any (любой);

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

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

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

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

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

На рисунке 5.7 представлены связи между атакующими действиями, использующими уязвимости соответствующей группы в рамках одного хоста.

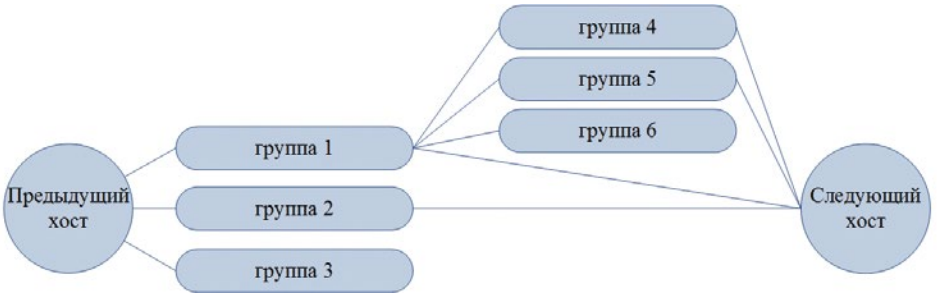


Рис. 5.7. Связи между группами уязвимостей

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

Ограничения модели: предполагается, что граф обладает свойством монотонности (то есть атакующий не возвращается и движется в направлении увеличения выигрыша) [130, 165]; граф не имеет циклов, так как для атакующего не имеет смысла повторное посещение уже посещенных узлов [132, 140] (это ведет к завышению оценок вероятности атаки); события компрометации узлов графа предполагаются независимыми.

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

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

Показатель *нормализованное количество атак с высоким уровнем риска, проходящих через хост HQNAH,* предлагается вычислять по формуле: $HQNAH = \frac{nh_i}{nh}$, где nh_i — количество атак с высоким уровнем риска, проходящих через хост *i*; *nh* — общее количество атак графа с высоким уровнем риска.

Алгоритмы вычисления основных показателей уровня графа атак. Для вычисления основных показателей уровня графа атак применяется модель атак (в форме Байесовского графа атак) и характеристики уязвимостей, применяемых при реализации атакующих действий, полученные из базы NVD [13]. В качестве основных были выбраны показатели: *сложность атакующего действия*; *сложность атаки*; *вероятность атаки*; *разрушительность атакующего действия*; *ущерб от атаки*.

Показатель *сложность атаки* вычисляется на основе индекса CVSS *сложность доступа* [97]. Пусть *AccessComplexity*(v_i) — *сложность доступа* уязвимости v_i, применяемой для реализации атакующего действия a_i. Тогда *сложность атакующего действия* a_i равна *AccessComplexity*(v_i). Сложность атаки *attackComplexity* определяется как max_i*AccessComplexity*(v_i), где $i \in [1, N]$, *N* — количество атакующих действий в атаке.

Показатель *ущерб от атаки aImpact* вычисляется на основе показателя *ущерб от атакующего действия* предыдущего уровня. Пусть *Impact*(a_i) — ущерб, наносимый ИС в результате успешного выполнения атакующего

действия a_i, $a_i \in A$, $i \in [1, N]$, где A — множество всех шагов атаки, N — количество шагов атаки. Тогда $aImpact = \max_i Impact(a_i)$.

Предлагаемый алгоритм определения *вероятности атаки* [202] использует и развивает предыдущие работы, применяющие байесовские графы атак [133, 138–141]. Отличия: метод формирования графа атак; метод вычисления локальных вероятностей компрометации узлов.

В работах [133, 138–141] предлагается применение следующего механизма для определения вероятности атаки на узел графа (байесовский метод определения вероятности события, при этом под вероятностью понимается степень доверия): (1) определение локальных вероятностей компрометации узлов графа (то есть вероятностей того, что атакующий сможет и проэксплуатирует уязвимость, соответствующую данному узлу, если все предусловия выполнены); (2) определение условных вероятностей компрометации узлов графа (дискретных локальных распределений условных вероятностей); (3) определение безусловных вероятностей компрометации узлов путем обхода графа.

В рамках разработанной методики оценивания защищенности локальные вероятности компрометации узлов определяются на основе индексов CVSS [97]. В [140] применяется показатель CVSS *Exploitability*. Данный показатель вычисляется в том числе на основе индекса CVSS *AccessVector* (*вектор доступа* к уязвимости). Предлагаемый в данном исследовании граф атак построен таким образом, что переход из состояния в состояние возможен только в случае наличия доступа к соответствующему узлу. Поэтому при определении локальных вероятностей успешной компрометации узла данный индекс учитывается только для корневых (входных) узлов графа. Таким образом, локальные вероятности успешной компрометации узла S_i, соответствующего атакующему действию a_i, определяются следующим образом: $p(a_i) = 2 \times AccessVector \times AccessComplexity \times Authentication$, если $S_i \in S_r$, где S_r — множество корневых (входных) узлов графа, *AccessComplexity* — *сложность доступа* к уязвимости по CVSS; *Authentication* — требуемая *аутентификация* по CVSS. В этом случае локальная вероятность успешной компрометации узла может принимать значения от 0,1 до 1 (в соответствии с возможными значениями индексов CVSS). Если $S_i \notin S_r : p(a_i) = 2 \times AccessComplexity \times Authentication$, в этом случае локальная вероятность успешной компрометации узла может принимать значения от 0,3 до 1. Вероятность того, что узел не будет скомпрометирован, определяется как 1-$p(a_i)$.

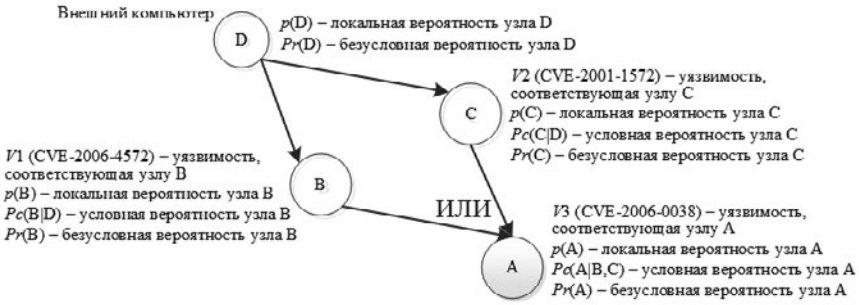
Для определения дискретных локальных распределений условных вероятностей *Pc* (то есть вероятностей компрометации узла с учетом различных комбинаций состояний его предков) необходимо учесть типы связей между узлами-предками. Обозначим $Pa(S_i)$ множество всех предков узла S_i, а функцию локального распределения условной вероятности $Pc(S_i|Pa(S_i))$. В [137] были предложены следующие две формулы. Для определения условной вероятности в случае связей типа «И» между узлами-предками (для успешной компрометации узла-потомка необходимо, чтобы все узлы-предки были скомпрометированы): $Pc\left(S_i \middle| Pa(S_i)\right) = \begin{cases} 0, & \exists S_i \in Pa(S_i) | S_i = 0 \\ p(S_i), & \text{иначе} \end{cases}$.

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

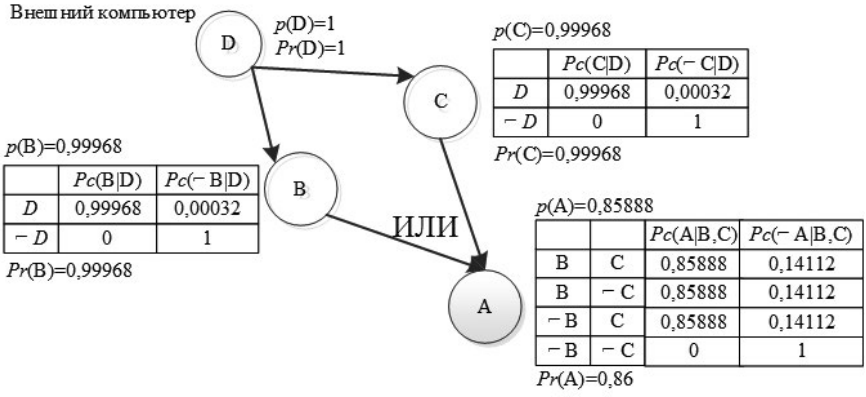
Для определения условных распределений вероятностей всех узлов применяется обратный обход графа в глубину, начиная с терминальных узлов (не имеющих потомков) и заканчивая узлами, доступными атакующему [195].

Безусловные вероятности компрометации узлов графа (*вероятности атаки*) определяются на основе локальных вероятностей и распределений условных вероятностей по формуле полной вероятности путем маргинализации по известным вероятностям: $\Pr(S_1, \dots, S_n) = \prod_{i=1}^{n} Pc\left(S_i \middle| Pa(S_i)\right)$, где S_i — i-й узел графа [195].

На рисунке 5.8, а представлен фрагмент графа атак, каждому узлу графа сопоставлена уязвимость и набор показателей (локальная вероятность, условная вероятность и безусловная вероятность). На рисунке 5.8, б для этого фрагмента приведены численные значения, рассчитанные в соответствии с формулами выше. Выходные данные работы алгоритма: итоговые значения *вероятностей атаки* для всех узлов графа.



а)



б)

Рис. 5.8. Фрагмент байесовского графа атак

Алгоритмы вычисления показателей уровня атакующего. Для определения показателей уровня атакующего применяется модель атакующего, которая задается как [166]: $A = (H_0, Sk, G)$, где H_0 ($H_0 \subseteq H$) — определяет хосты, к которым имеет доступ атакующий до проведения атак и уровень привилегий на них, G — содержит цели атакующего ($G = <H \times I, H \times R>$, где R и I определяют результат атакующего действия: получение прав доступа и/или воздействие на информацию), Sk — определяет уровень навыков атакующего.

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

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

Алгоритмы вычисления показателей уровня событий. Под событием безопасности будем понимать обработанное событие, поступающее от SIEM-системы. Например: получение нелегитимных привилегий на хосте; нарушение конфиденциальности хоста. В рамках проекта MASSIF [87] события безопасности, прежде чем поступить на вход компонента оценки защищенности, проходят через систему корреляции и обработки событий (рисунок 1.7), которая извлекает так называемое «сырое» событие (т. е. текстовое сообщение, содержащее все данные исходного сообщения) из сообщения, формирует сообщение необходимого формата на основе извлеченных данных и на основе правил корреляции формирует события более высокого уровня.

На уровне событий состояние графа атак S_i меняется ввиду поступления нового события ev_i, определяющего, что один из узлов графа атак перешел в состояние «скомпрометирован». Подобная формализация предложена в [210] для моделирования распространения ущерба в графах зависимостей сервисов при поступлении внешнего события об атаке от СОВ. Новое состояние графа S_{i+1} характеризуется тем, что оценка вероятности компрометации соответствующего узла изменилась (рисунок 5.9). В результате меняются оценки вероятности компрометации и рисков для последующих узлов графа атак.

Рис. 5.9. Изменение состояния системы в результате поступления события

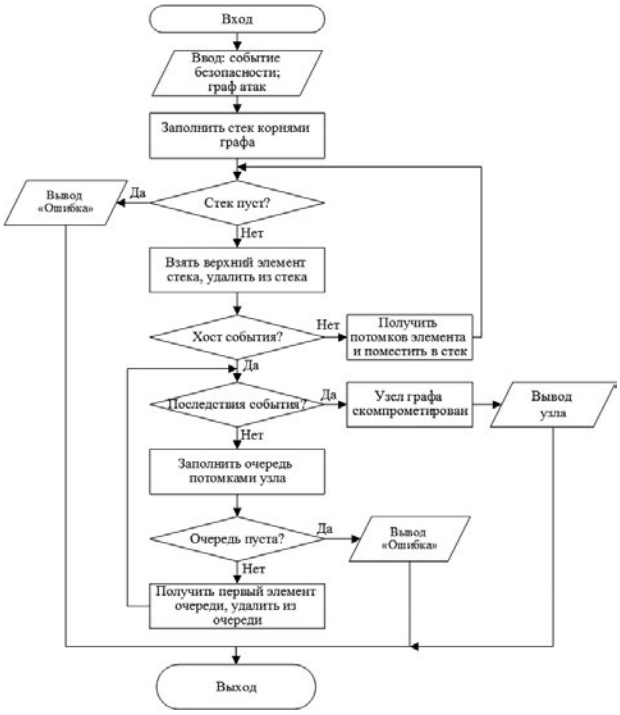


Рис. 5.10. Алгоритм отображения события безопасности на граф атакующих действий

Согласно классификации [211] для описания события безопасности используются категории: атакующий (кто проводит атаку); метод атаки (методы эксплуатации уязвимости); уязвимость (тип использованной уязвимости); действие (шаги для достижения результата); объект атаки (на что направлена атака); результат атаки (последствия инцидента); цели (цели атаки). В исследовании для отображения события безопасности на граф атакующих действий используются поля: объект атаки (учетная запись, процесс, данные, компонент, компьютер, система или сеть); результат атаки (эскалация привилегий, раскрытие информации, разрушение информации, отказ в обслуживании, кража ресурсов). При наличии, можно учитывать дополнительную информацию: атакующий (кто проводит атаку); цели (цели атаки). Модель события задается следующим образом: $E = (Ti, H, Te)$, где Ti — время произошедшего изменения (в данном исследовании рассматриваются дискретные моменты времени); H — модель хоста, на котором произошло событие; Te — тип события ($Te = \{R, I, \text{other}\}$, где R — получение прав доступа, I — воздействие на информацию, т.е. нарушение конфиденциальности/целостности/доступности).

Вычисление вспомогательных показателей уровня событий. Показатель *нормированное количество инцидентов на хосте* отображает количество инцидентов на хосте по отношению к общему количеству инцидентов в системе: $NNHI = \frac{NI_i}{NI}$, где NI_i — количество событий безопасности, поступивших от SIEM-системы для i-го хоста (*количество инцидентов на хосте*);

NI — общее количество событий безопасности, поступивших от SIEM-системы по всем хостам сети (*количество инцидентов в системе*). Показатель измеряется в процентах и может изменяться во времени.

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

Алгоритмы вычисления основных показателей уровня событий. На данном уровне вычисляются значения следующих основных показателей: *динамический уровень навыков атакующего*; *динамическая вероятность атаки*. Входные данные для вычисления показателей: входные данные и показатели предыдущих уровней; модель события.

Алгоритм вычисления основных показателей уровня событий включает этапы [202]: (1) отображение события на граф атак; (2) переопределение уровня навыков атакующего; (3) переопределение вероятности атаки. Схема алгоритма отображения события безопасности на граф атакующих действий представлена на рисунке 5.10. Алгоритм заключается в поиске узла графа по хосту, которому соответствует событие безопасности и последствиям атакующего действия, сгенерировавшего событие. Результатом работы алгоритма является позиция атакующего на графе атак.

Динамический уровень навыков атакующего определяется на основе шагов:

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

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

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

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

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

Результат работы алгоритма: наиболее вероятный путь до позиции атакующего на графе атак; *динамическая вероятность атаки* для узла, соответствующего позиции атакующего на графе, и его предков; *динамический уровень навыков атакующего* и точность определения уровня навыков атакующего.

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

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

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

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

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

Результатом работы алгоритма является *динамическая вероятность атаки*.

Алгоритмы вычисления показателей уровня выбора контрмер. Уровень защищенности в статическом режиме и распространение атаки в динамическом режиме зависит от того, реализованы или нет защитные меры M_i. Защитные меры влияют на изменение состояния графа атак при реализации атакующего действия (рисунок 5.11).

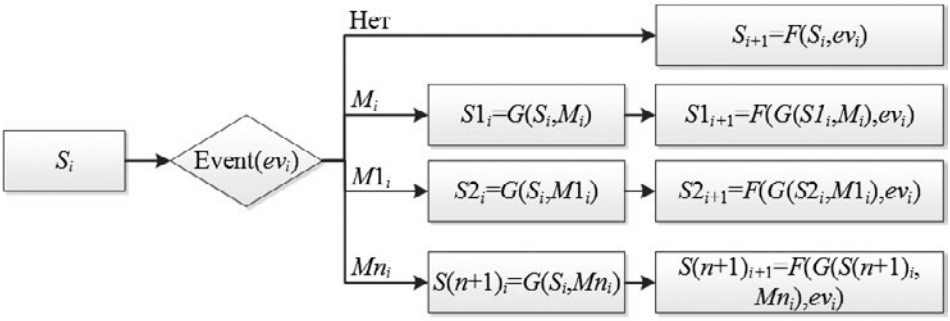


Рис. 5.11. Влияние защитных мер на состояние графа атак [210]

Для вычисления показателей уровня выбора контрмер вводится модель контрмеры [201, 204]. Для создания модели контрмер использовались стандарты протокола SCAP: CRE [128] и ERI [129]. Модель контрмеры определяет: характеристики контрмеры, необходимые для ее включения в методику выбора контрмер; возможные значения характеристик контрмеры; связь характеристик контрмеры с методикой выбора контрмер.

Предлагаемая концептуальная модель контрмеры включает поля (рисунок 5.12):

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

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

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

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

Поля:

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

Пример значений полей:

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

Рис. 5.12. Поля концептуальной модели контрмеры

В данном исследовании выделяются статический и динамический режимы работы. В статическом режиме контрмеры включают различные инструменты, которые позволяют снизить уровень риска. В динамическом режиме рассматриваются контрмеры, которые могут предотвратить распространение атаки вглубь ИС. Например, в статическом режиме в систему может быть добавлен межсетевой экран, а в динамическом режиме он может применяться для блокировки подозрительных учетных записей. Данное разделение основано на предположении, что в динамическом режиме работы системы нет времени на развертывание дополнительных инструментов защиты. Примеры контрмер согласно [39]:

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

Логическое управление и аудит доступа. В статическом режиме может быть реализовано путем мониторинга сети с использованием продуктов FreeNATS [212] или NetCrunch [213]. В динамическом режиме данные продукты могут использоваться для активации правил аномального поведения.

Обнаружение и предотвращение вторжений. В статическом режиме для обнаружения и предотвращения вторжений могут использоваться СОВ и системы предотвращения вторжений (СПВ). В динамическом режиме они позволяют реализовать контрмеры: активация СОВ и СПВ в стратегических местах, смена портов соединения. Примеры СОВ на хостах: Intrust [214], Snort [215]. Пример СПВ: Cisco 500 Se [216].

Предохранение от злонамеренного кода. Для предохранения от злонамеренного кода в статическом режиме могут использоваться сканеры. В динамическом режиме они позволяют реализовать удаление злонамеренного кода. Пример антивирусного сканера: Kaspersky [217].

Управление безопасностью сети (планирование, эксплуатация и администрирование сетей). Для обеспечения безопасности сети в статическом режиме могут использоваться межсетевые переходы безопасности, виртуальные частные сети, СОВ и СПВ, мониторинг сети, межсетевые экраны. В динамическом режиме СОВ и СПВ позволяют реализовать контрмеры: активация СОВ в стратегических местах, смена портов соединения. Мониторинг сети позволяет активировать правила аномального поведения. Межсетевые экраны позволяют реализовать контрмеры: блокировка подозрительных учетных записей, временная деактивация учетных записей пользователей (на 24, 48 или 72 часа). Примеры межсетевых экранов: Comodo [218], Endian [219]. Кроме того, можно усилить защищенность сети путем обновления ПО (и удаления уязвимостей).

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

Выбор данных контрмер обусловлен тем, что они относятся к специальным защитным мерам, применяемым в системах, для которых требуется детальный анализ рисков (для которых разрабатывались рассматриваемые методики оценивания защищенности и выбора контрмер).

В таблице 5.2 представлена связь контрмер с угрозами различным свойствам безопасности.

В таблице применяются обозначения: S — применяется в статическом режиме, D — применяется в динамическом режиме, SD — применяется в обоих режимах; C — конфиденциальность, I — целостность, A — доступность. Классификация основана на стандарте [39].

При создании модели учитывалось применение графа атак и графа зависимостей сервисов для оценивания защищенности. Граф зависимостей сервисов применяется при определении побочного ущерба, наносимого в ре-

зультате реализации контрмер. Кроме того, реализация контрмеры влияет на переходы состояний и, соответственно, изменяет граф атак.

Таблица 5.2

Примеры угроз, свойств безопасности и контрмер

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

Очевидно, что контрмера может повлиять на каждый из элементов графа атак (узел и дуга) тремя способами: удаление, добавление, изменение (например, вероятности атак) (рисунок 5.13). Зеленые стрелки на рисунке 5.13 обозначают добавление, красные — удаление, синие — изменение. Жирные стрелки используются для связи статических контрмер и влияния на граф атак. Пунктирные и сплошные стрелки показаны на рисунке 5.13 для выде-

ления путей, соответствующих определенным контрмерам, например, открытие порта обуславливает добавление дуги, но не узла.

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

Основные выбранные показатели уровня выбора контрмер: *уровень побочного ущерба*; *эффективность контрмеры*; *стоимость контрмеры*.

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

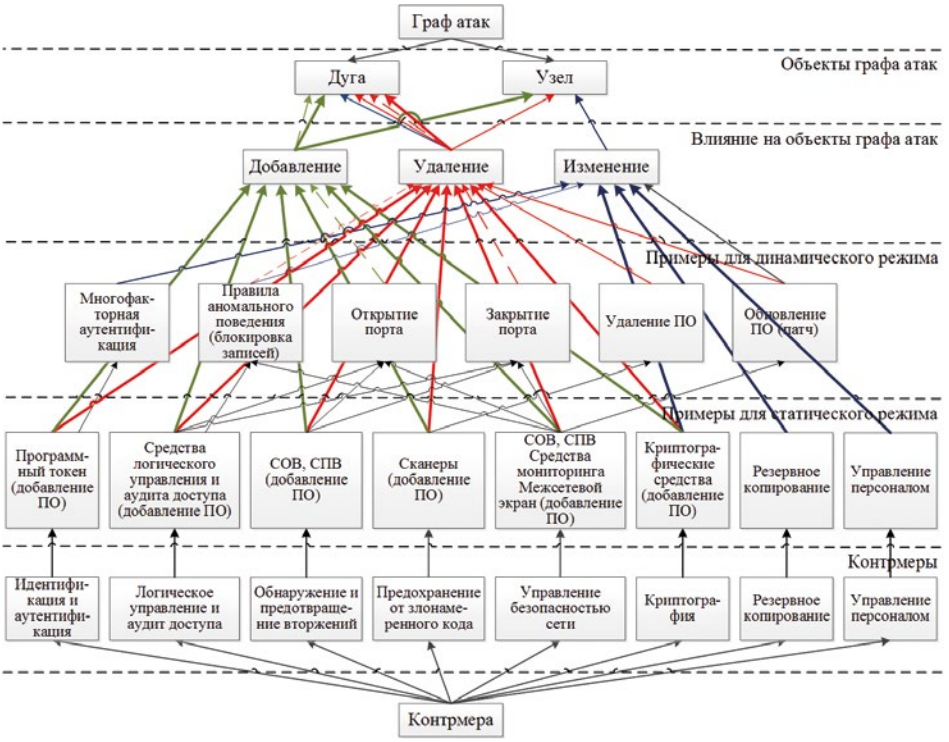


Рис. 5.13. Зависимости между контрмерами и объектами графа атак

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

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

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

Под риском понимается комбинация факторов вероятности возникновения инцидента (или проведения атаки) и его разрушительного воздействия [36]. Входные данные алгоритма определения уровня риска: модели входных данных соответствующего уровня, в том числе модель ИС и модель атак; показатели защищенности соответствующего уровня в том числе *ущерб от атаки* и *вероятность атаки*. Алгоритм включает этапы: сбор входных данных; определение уровня вычислений; вычисление значения риска. Выходные данные алгоритма: *риск* соответствующего уровня.

Алгоритм вычисления уровня риска на уровне инфраструктуры. Для определения уровня риска на уровне инфраструктуры предлагается использовать оценки CVSS для уязвимостей. Уравнения CVSS позволяют учитывать ущерб, наносимый уязвимостью, и вероятность ее использования, что удовлетворяет определению риска, данному в [39]. Согласно разработанной методике оценивания защищенности ИС, на уровне инфраструктуры не учитывается последовательное применение нескольких уязвимостей для реализации многошаговых атак, поэтому отсутствие корреляции между уязвимостями в уравнениях CVSS не является ограничением. Уровень риска предлагается определять на основе модифицированного контекстного уравнения CVSS, так как оно позволяет учитывать связь между оценкой уязвимости и критичностью активов. Контекстное уравнение [98]:

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

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

TemporalScore — временная оценка по CVSS;

CollateralDamagePotential — потенциал побочного ущерба при эксплуатации уязвимости;

TargetDistribution — плотность целей.

AdjustedImpact определяется по формуле:

$$AdjustedImpact = \min(10{,}10.41 \times (1 - (1 - ConfImpact \times ConfReq) \times (1 - IntegImpact \times IntegReq) \times (1 - Availmpact \times AvailReq))),$$

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

ConfReq, *IntegReq*, *AvailReq* — требования безопасности.

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

Тогда уравнение принимает вид: *Risk*=round_to_1_ decimal(*AdjustedTemporal*).

Временная оценка определяется на основе уравнения [98]:

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

где *Exploitability* — возможность использования уязвимости;

RemediationLevel — уровень исправления уязвимости;

ReportConfidence — степень достоверности отчета об уязвимости.

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

$$AdjustedBase = round\_to\_1\_decimal(((0.6 \times AdjustedImpact) + (0.4 \times Exploitability) - 1.5) \times f(AdjustedImpact)),$$

где $f(AdjustedImpact) = \begin{cases} 0, \text{если } AdjustedImpact = 0 \\ 1.176, \text{если } AdjustedImpact \neq 0 \end{cases}$;

Exploitability — возможность использования уязвимости.

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

При подстановке получаем:

$$Risk = round\_to\_1\_decimal(((0.6 \times AdjustedImpact) + (0.4 \times Exploitability) - 1.5) \times f(AdjustedImpact).$$

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

$$AdjustedImpact = \min(10, 10.41 \times (1 - (1 - ConfImpact \times Criticality(c)) \times (1 - IntegImpact \times Criticality(i)) \times (1 - AvailImpact \times Criticality(a)))),$$

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

Возможные значения показателей критичности и их преобразование для применения в уравнении оценки риска приведены в таблице 5.3. Таким образом, риск может принимать значения от 0 до 10.

Таблица 5.3

Преобразование оценок критичности актива для применения в уравнении оценки риска

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

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

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

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

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

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

Алгоритм вычисления индекса выбора контрмеры. Показатель *индекс выбора контрмеры CI* определяется на основе выигрыша в результате реализации контрмеры и затрат на ее реализацию (а именно, стоимости реализации контрмеры и побочного ущерба от ее реализации) [210]. Для этого определяются возможные потери до ($Risk_a$) и после ($Risk_b$) реализации контрмеры. Выигрыш определяется как разница между $Risk_a$ и $Risk_b$: $Benefit=Risk_a — Risk_b$, где $Risk_a$ являются суммой риска по всем узлам, за-

трагиваемым контрмерой до ее реализации, а $Risk_b$ — после ее реализации. При этом выигрыш должен стремиться к максимально возможному значению, а затраты — к минимальному. Поэтому *индекс выбора контрмеры CI* определим как:

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

где *CD — побочный ущерб*,

CC — стоимость контрмеры.

При выборе контрмер необходимо максимизировать данный показатель.

Анализ индекса выбора контрмеры. Рассмотрим граничные значения параметров, применяемых для вычисления данного показателя. В случае если *Benefit*<=0, показатель принимает значение меньшее или равное 0, поэтому на значение данного показателя накладывается ограничение *CI*>0, так как отсутствие выигрыша не имеет смысла, а отрицательное значение показателя говорит о том, что реализация контрмеры приведет к еще большим потерям, чем атака. Не учитывается ситуация, когда стоимость реализации контрмеры (*CD*+*CC*) равна 0. В дальнейшем эту проблему можно будет решить, например, на основе идеи, реализованной в [189], путем учета стоимости инфраструктуры. Кроме того, необходимо учитывать, что в случае, если выигрыш и затраты намного меньше единицы (стремятся к минимальным значениям), и выигрыш и затраты намного больше единицы (стремятся к максимальным значениям), индекс может принимать одинаковые значения. В этом случае можно задавать дополнительное условие (что важнее — максимально снизить риск или минимизировать затраты).

5.3. Методика выбора контрмер

Была разработана методика выбора контрмер на основе предложенного комплекса показателей защищенности и графов атак [201, 204]. Методика выбора контрмер включает модель контрмеры, связь модели контрмер и графа атак, алгоритм вычисления *индекса выбора контрмеры*. Выделяются статический и динамический режимы работы методики выбора контрмер. Методика статического режима применяется на этапах проектирования и развертывания системы, методика динамического режима — на этапе эксплуатации. В статическом режиме выбираются контрмеры, которые позволят повысить общий уровень защищенности системы. В динамическом — меры, которые позволят остановить атаку. Примеры контрмер: патч для уязвимости (информацию можно взять, например, из базы xForce [220], которая содержит временные оценки CVSS, в том числе *уровень исправления*, который определяет наличие патча для уязвимости); удаление уязвимого ПО; закрытие порта; добавление дополнительных защитных средств (например, межсетевой экран или антивирус).

Списки контрмер, оценки их эффективности и стоимость определяются экспертами.

Методика выбора контрмер в статическом режиме на уровне инфраструктуры. Входные данные для методики выбора контрмер на уровне инфраструктуры: набор доступных контрмер; модель анализируемой ИС

(включая программно-аппаратное обеспечение и его уязвимости, дающие высокий уровень риска); показатели топологического уровня (включая риск уязвимостей, ПО и хостов; ущерб, наносимый эксплуатацией уязвимостей, по свойствам конфиденциальности, целостности и доступности).

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

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

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

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

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

   а) Для всех уязвимостей экземпляра ПО, формирующих высокую оценку риска, определяется показатель CVSS у*ровень исправления* (*RemediationLevel*). В случае наличия исправления считается *ин-*

декс выбора контрмеры для контрмеры «исправление уязвимости». При этом риск пересчитывается на основе временного уравнения CVSS *TemporalScore*=round_to_1_decimal(*BaseScore*r×*RemediationLevel*), где функция round_to_1_decimal выполняет округление аргумента до одного знака после запятой. Суммарный показатель *индекса выбора контрмеры* по контрмере «исправление уязвимости» вычисляется на основе максимального *риска* по уязвимостям и суммарной *стоимости*.

б) Если все уязвимости экземпляра ПО рассмотрены, для экземпляра ПО выбирается максимальный *индекс выбора контрмеры* (и соответствующие контрмеры) из контрмер по уровню ПО и уровню уязвимостей и принимается за *И3*.

4) Вычисляется общий *индекс выбора контрмеры* по всем экземплярам ПО, с учетом контрмер, выбранных для каждого экземпляра, выбором максимального риска среди всех *И3* и суммарной стоимости и принимается за *И3*.

5) Вычисляется общий *индекс выбора контрмеры* по свойству выбором максимального риска из *И2* и *И3* и суммарной стоимости. Обозначим его *И4*.

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

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

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

Методика выбора контрмер на уровне графа атак реализуется в несколько этапов (алгоритм работы методики представлен на рисунке 5.14):

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

2) Определить узлы графа, представляющие наибольший риск. Для этого применяется алгоритм, предложенный в [149], адаптированный для графа атак, применяемого в данном исследовании (исходный алгоритм работает для отдельных уязвимостей, в данном исследовании он применяется для групп уязвимостей, соответствующих узлам графа атак).

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

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

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

Рис. 5.14. Алгоритм выбора контрмер на уровне графа атак

Методика выбора контрмер в динамическом режиме. Методика *уровня событий* (рисунок 5.15) отличается тем, что основной целью является предотвращение обнаруженной атаки, а не повышение общего уровня защищенности ИС. На уровне событий контрмеры реализуются в зависимости от текущих и будущих (спрогнозированных) шагов атакующего. При этом учитывается «глубина графа до критичного ресурса», которая определяется как количество узлов графа до актива с высоким уровнем критичности. Если данная глубина превышает определенное значение, то система ждет нового события для уточнения своих оценок, если глубина меньше определенного значения, система предлагает контрмеру на основе имеющихся данных со степенью точности, соответствующей количеству уже выявленных релевантных событий.

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

Методика включает следующие основные этапы (рисунок 5.15):

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

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

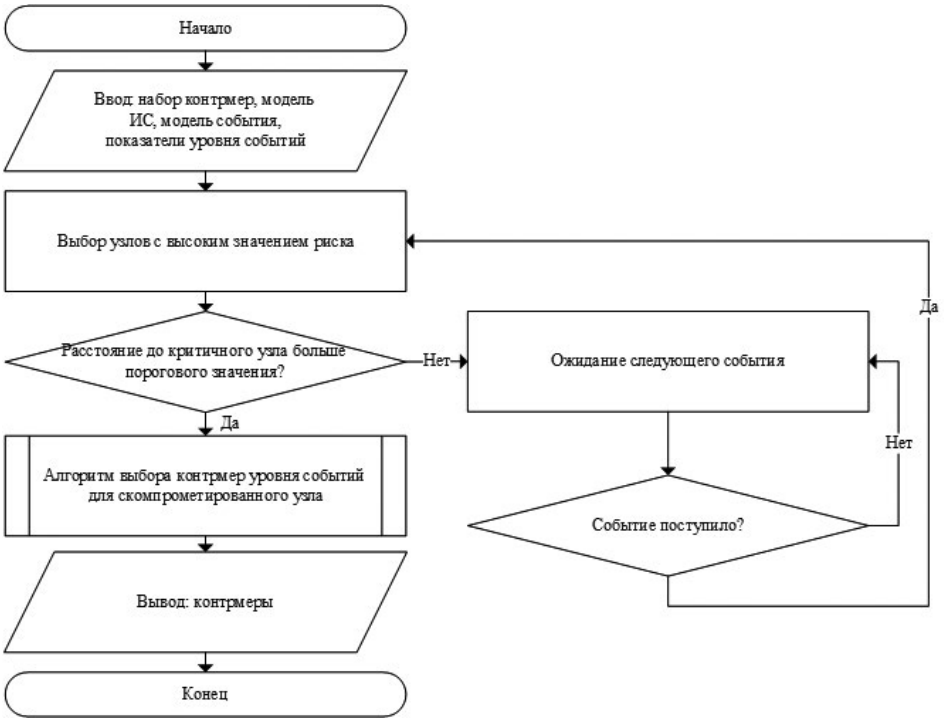


Рис. 5.15. Методика выбора контрмер на уровне событий

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

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

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

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

5.4. Вариант архитектуры и реализация средств оценивания защищенности и выбора контрмер

Разработанные методики и алгоритмы реализованы в рамках системы оценивания защищенности и выбора контрмер (СОЗВК) для компьютер-

ных систем и сетей, которая является частью системы анализа защищенности на основе графов атак и графов зависимости сервисов, реализованной в рамках SIEM-системы, разработанной в проекте MASSIF [87]. СОЗВК реализует следующие функции: (1) получение адекватной и актуальной оценки защищенности ИС на основе доступных входных данных в статическом и динамическом режимах работы; (2) учет характеристик атакующего; (3) учет взаимосвязей между сервисами ИС для тщательного учета возможного распространения ущерба в случае успешной реализации атак, или побочного ущерба при реализации защитных мер, в статическом и динамическом режимах работы; (4) учет стоимостных характеристик атак и контрмер для того, чтобы определить выигрыш в случае реализации контрмер в статическом и динамическом режимах работы; (5) автоматизация процесса представления и обработки данных, применяемых для оценивания защищенности и выбора контрмер, в статическом и динамическом режимах работы; (6) выбор наиболее адекватного решения по реагированию с учетом стоимостных требований в статическом и динамическом режимах работы; (7) выявление слабых мест ИС в статическом режиме работы; (8) выявление возможных атак на ИС и получение набора показателей защищенности, характеризующего их, в статическом режиме работы; (9) выбор средств защиты для повышения уровня защищенности системы в статическом режиме работы; (10) учет событий безопасности, происходящих в ИС, и переоценка ситуации по защищенности в соответствии с полученной информацией в динамическом режиме работы; (11) интеграция с SIEM-системами в динамическом режиме работы; (12) выбор контрмер для предотвращения развивающейся атаки в динамическом режиме работы.

Вариант архитектуры системы представлен на рисунке 5.16. Компонент обработки данных получает входные данные от администратора, компонента сбора информации (который получает входные данные от сенсоров, сетевых сканеров, хостовых программных агентов, SIEM-системы, и обрабатывает получаемые данные), компонента моделирования атак и генерирует модели входных данных. Полученные модели применяются как входные данные для компонента оценивания защищенности. Компонент оценивания защищенности включает набор функций, реализующих методики вычисления показателей защищенности уровня, соответствующего получаемым моделям, и методику оценивания защищенности. В случае поступления новых данных показатели пересчитываются. Далее полученные показатели применяются для выбора контрмер.

Выходные данные системы включают уровень защищенности компьютерной системы или сети, вычисленные показатели защищенности и набор контрмер. Выходные данные передаются для отображения системе визуализации. Ниже подробнее рассмотрены основные компоненты СОЗВК.

Система визуализации, *компонент моделирования атак* и *база данных* являются внешними системами по отношению к СОЗВК. *Система визуализации* позволяет пользователю управлять системой, задавать входные данные и просматривать результаты работы системы [221]. *Компонент моделирования атак* [166] формирует исходное дерево атак на основе которого генерируется Байесовский граф атак, применяемый в СОЗВК для оценива-

ния защищенности. В *базе данных* [166] хранятся конфигурации анализируемой ИС, реализуемая в сети политика безопасности (общие правила функционирования сети), события, происходившие в системе (зафиксированные действия атакующего), возможные и реализованные контрмеры и результаты их внедрения. *Компонент обработки данных* объединяет функции формирования моделей на основе получаемых входных данных: (1) генератор графа зависимостей сервисов преобразует зависимости между сервисами в граф; (2) генератор графа атак преобразует дерево атак в Байесовский граф атак. *Компонент оценивания защищенности* содержит функции расчета предложенных в исследовании показателей защищенности. Он включает функцию анализа имеющихся входных данных, необходимую для определения уровня вычислений, и функцию сравнительной оценки риска, необходимую для определения уровня защищенности системы. *Компонент выбора контрмер* содержит функции выбора контрмер статического и динамического режимов и компонент анализа входных данных (для определения уровня вычислений). *Генератор сценариев атак* и *компонент оценки эффективности* разработаны для оценки эффективности разработанной СОЗВК. *Генератор сценариев атак* содержит функции генерации последовательностей атак, приближенных к реальным действиям атакующего в системе, и функции генерации последовательностей соответствующих им событий.

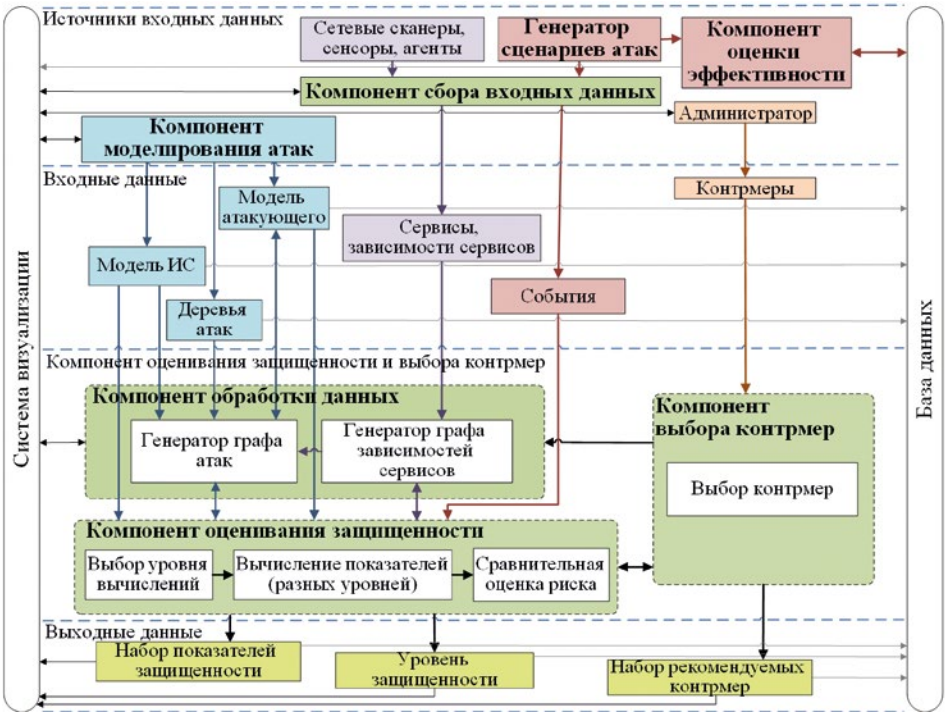


Рис. 5.16. Архитектура системы оценивания защищенности и выбора контрмер

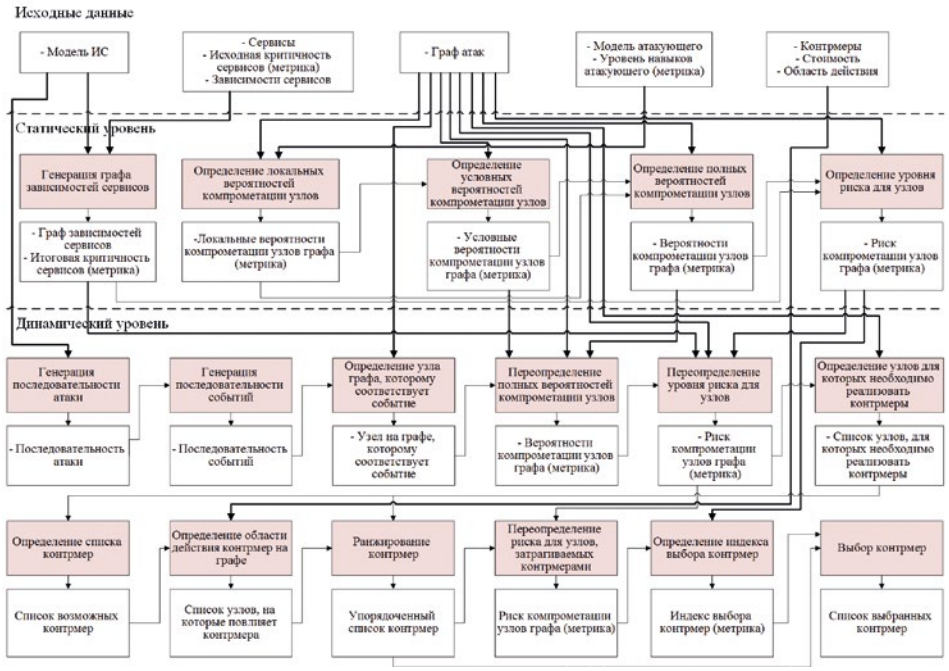


Рис. 5.17. Функциональная схема прототипа системы оценивания защищенности и выбора контрмер

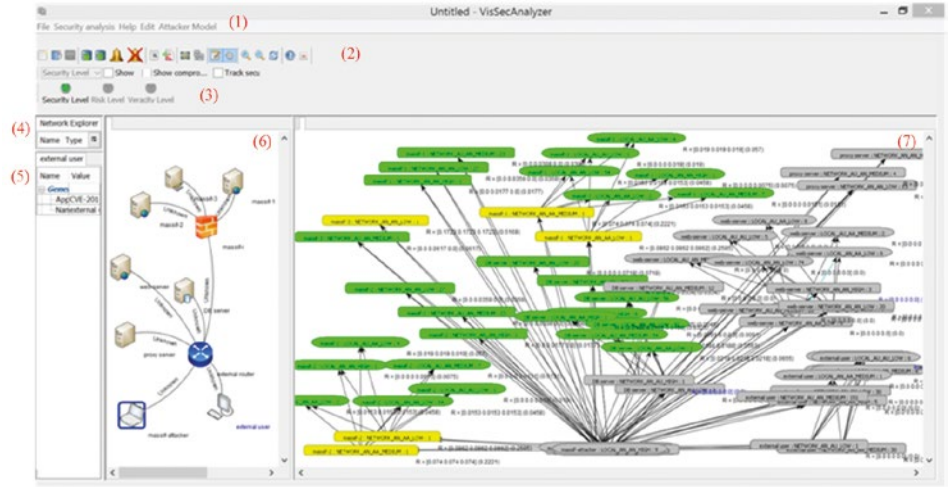


Рис. 5.18. Графический интерфейс пользователя программного прототипа СОЗВК

Разработанные методики реализованы в рамках программного прототипа СОЗВК. Функциональная схема прототипа СОЗВК представлена на рисунке 5.17. Прототип был реализован на языке Java с использованием принципов объектно-ориентированного программирования, на ОС Microsoft Windows, Intel Core i5 CPU и 12 GB ОЗУ. Графический интерфейс пользо-

вателя программного прототипа СОЗВК представлен на рисунке 5.18. Графический интерфейс пользователя содержит: (1) главное меню; (2) панель управления; (3) панель графического представления уровня защищенности, вычисляемого с использованием методики оценивания защищенности; (4) диалог доступа к сети (Network Explorer); (5) диалог свойств сети (Property Explorer); (6) окно представления конфигурации ИС; (7) окно представления графа атак.

Выводы по главе 5

Разработанная в СПб ФИЦ РАН система оценивания защищенности и выбора контрмер реализует оригинальные методики оценивания защищенности и принятия решений, основанные на графах атак, графах зависимостей сервисов и комплексе показателей защищенности. Применение байесовского графа атак позволяет моделировать возможные сценарии атак и оценивать риски информационной безопасности с учетом вероятности реализации атак и ущерба от их реализации, а также с учетом происходящих в системе событий безопасности. Применение графа зависимостей сервисов позволяет учесть распространение ущерба в системе в случае успешной реализации атак или контрмер. Разработанная СОЗВК предназначена для разнородных, распределенных, открытых компьютерных систем с повышенными требованиями к кибербезопасности. Она позволяет оценивать защищенность компьютерных систем и сетей в статическом и динамическом режимах, прогнозировать развитие атаки и выбирать контрмеры, снижающие ущерб от кибератак.

Заключение

Противодействие киберугрозам и управление кибербезопасностью является одним из приоритетных направлений научно-технологического развития РФ. Значимость решения данной задачи обусловлена существенным влиянием, которое оказывают результаты ее решения в целом на национальную безопасность страны.

Важным элементом эффективного противодействия киберугрозам является сбор, обработка и анализ информации, а также формирование значимых для принятия решений количественных показателей защищенности.

В монографии детально рассмотрены существующие стандарты и руководящие документы, методики и практические решения в области оценивания защищенности и выбора контрмер для управления кибербезопасностью, а также применяемые показатели защищенности.

Основное внимание уделено графовым моделям, представляющим интерес и возможности для проактивного оценивания защищенности и выбора контрмер. Хотя существуют различные стандарты оценивания защищенности и выбора контрмер, соответствующие методики и показатели защищенности, системы мониторинга безопасности и управления инцидентами, текущая статистика в области киберпреступлений указывает на то, что применение традиционных средств и систем защиты информации является недостаточным.

В работе описаны существующие проблемы оценивания защищенности и выбора контрмер при управлении кибербезопасностью, в недостаточной степени исследованные с точки зрения получения фундаментальных теоретических и практических результатов в области защиты информации.

Ввиду изменчивого характера предметной области — постоянного роста сложности защищаемых систем, в том числе разнородности применяемых устройств, разнородности и объемов данных безопасности — актуальным направлением решения проблем управления кибербезопасностью представляется интеллектуализация сервисов защиты информации, в первую очередь, сервисов, которые осуществляют оценивание защищенности и управление защитой. Хотя в этой области исследователями предложено большое количество решений, они по большей части не находят должного практического применения.

В монографии описаны разработанные авторами методики и средство оценивания защищенности и выбора контрмер на основе графов атак и зависимостей сервисов, представляющие собой интеллектуальные сервисы управления кибербезопасностью и показавшие хорошие результаты в рамках проведенных экспериментов.

Дальнейшее развитие предложенных методик связано с применением методов интеллектуального анализа больших данных в целях усовершенствования анализа и прогнозирования кибератак, профилирования атакующих, выбора контрмер с точки зрения точности и производительности.

Результаты, рассмотренные в монографии, могут использоваться при разработке и построении систем управления кибербезопасностью, а также позволят исследователям и разработчикам систем управления кибербезопасностью сосредоточиться на существующих научных проблемах в этой области.

Список литературы и электронных ресурсов

1. **Костина Н. В.** Основные этапы развития теории риска / Н. В. Костина // Тр. Псков. политех. ин-та. — Псков, 2010. — № 14.2. — С. 194–199.

2. **Костина Н. В.** Истоки возникновения и методические основы анализа предпринимательского риска / Н. В. Костина // Вест. Южно-Урал. гос. ун-та. Серия «Экономика и менеджмент». — 2012. — № 22. — С.147–151.

3. **Lynett M.** A History of Information Security From Past to Present [Электронный ресурс]. — 2015. — Режим доступа: https://blog.mesltd.ca/a-history-of-information-security-from-past-to-present (по состоянию на 27.04.2021).

4. **Завгородний В. И.** Системное управление информационными рисками. Выбор механизмов защиты // Проблемы управления. — 2009. — № 1. — С. 53–58.

5. **ГОСТ Р ИСО/МЭК 27005–2010**. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности [Текст]. — Введ. 2010–11–30. — М.: Стандартинформ, 2011. — 47 с.

6. **ГОСТ Р ИСО/МЭК 13335-5-2006**. Информационная технология. Методы и средства обеспечения безопасности. Часть 5. Руководство по менеджменту безопасности сети [Текст]. — Введ. 2007–06–01. — М.: Стандартинформ, 2007. — 22 с.

7. **Астахов А.** Искусство управления информационными рисками [Текст] / А. М. Астахов. — М.: ДМК Пресс, 2010. — 312 с.

8. **Котенко И. В.** Анализ протокола автоматизации управления данными безопасности SCAP [Текст] / И. В. Котенко, Е. В. Дойникова // Защита информации. Инсайд. — СПб, 2012. — № 2. — С. 56–63.

9. Common Vulnerabilities and Exposures (CVE) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://cve.mitre.org (по состоянию на 27.04.2021).

10. Common Attack Pattern Enumeration and Classification (CAPEC) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://capec.mitre.org/ (по состоянию на 27.04.2021).

11. **Котенко И. В.** Общее перечисление и классификация шаблонов атак (CAPEC): описание и примеры применения [Текст] / И. В. Котенко, Е. В. Дойникова, А. А. Чечулин // Защита информации. Инсайд. — СПб., 2012. — № 4. — С. 54–66.

12. Common Platform Enumeration (CPE) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://nvd.nist.gov/products/cpe (по состоянию на 27.04.2021).

13. NVD website [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://nvd.nist.gov/ (по состоянию на 27.04.2021).

14. MITRE ATT&CK website [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://attack.mitre.org/ (по состоянию на 27.04.2021).

15. Global Threat Intelligence Report / NTTGroup; NTTInnovationInstitute 1 LLC. — 2014. — 67 p.

16. Group-IB. Hi-Tech Crime Trends 2020/2021. — Ноябрь 2020.

17. Нормативные правовые акты, организационно-распорядительные документы, нормативные и методические документы и подготовленные проекты документов по технической защите информации [Электронный ресурс] : ФСТЭК России. Официальный сайт. — Электрон. текстовые данные. — [Б. м. : б. и.], 2021. — Режим доступа: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty/113-gosudarstvennye-standarty/377-gosudarstvennye-standarty (по состоянию на 28.04.2021).

18. **ГОСТ Р ИСО/МЭК 27000–2012**. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Общий обзор и терминология [Текст]. — Введ. 2012–11–15. — М.: Стандартинформ, 2014. — 22 с.

19. **ГОСТ Р ИСО/МЭК 15408-1-2012**. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель [Текст]. — Введ. 2013–12–01. — М.: Стандартинформ, 2014. — 56 с.

20. **ГОСТ Р ИСО/МЭК 15408-2-2013**. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2. Функциональные компоненты безопасности [Текст]. — Введ. 2014–09–01. — М.: Стандартинформ, 2014. — 336 с.

21. **ГОСТ Р ИСО/МЭК 15408-3-2013**. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности [Текст]. — Введ. 2014–09–01. — М.: Стандартинформ, 2014. — 274 с.

22. **ГОСТ Р 54581–2011**. Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 1. Обзор и основы [Текст]. — Введ. 2012–07–01. — М.: Стандартинформ, 2018. — 27 с.

23. **ГОСТ Р 54582–2011**. Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 2. Методы доверия [Текст]. — Введ. 2012–12–01. — М.: Стандартинформ, 2019. — 52 с.

24. **ГОСТ Р 54583–2011**. Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 3. Анализ методов доверия [Текст]. — Введ. 2012–12–01. — М.: Стандартинформ, 2013. — 54 с.

25. **ГОСТ Р 57628–2017**. Информационная технология. Методы и средства обеспечения безопасности. Руководство по разработке профилей защиты и заданий по безопасности [Текст]. — Введ. 2018–01–01. — М.: Стандартинформ, 2017. — 102 с.

26. **ГОСТ Р ИСО/МЭК 18045–2013**. Информационная технология. Методы и средства обеспечения безопасности. Методология оценки безопасности информационных технологий [Текст]. — Введ. 2014–07–01. — М.: Стандартинформ, 2014. — 249 с.

27. **ГОСТ Р ИСО/МЭК 19791–2008**. Информационная технология. Методы и средства обеспечения безопасности. Оценка безопасности автоматизированных систем [Текст]. — Введ. 2009–10–01. — М.: Стандартинформ, 2010. — 126 с.

28. **ГОСТ Р ИСО/МЭК 58143–2018**. Информационная технология. Методы и средства обеспечения безопасности. Детализация анализа уязвимостей программного обеспечения в соответствии с ГОСТ Р ИСО/МЭК 15408 и ГОСТ Р ИСО/МЭК 18045 Часть 2 Тестирование проникновения [Текст]. — Введ. 2018–05–24. — М.: Стандартинформ, 2018. — 22 с.

29. **ГОСТ Р ИСО/МЭК 27031–2012**. Информационная технология. Методы и средства обеспечения безопасности. Руководство по готовности информационно-коммуникационных технологий к обеспечению непрерывности бизнеса [Текст]. — Введ. 2014–01–01. — М.: Стандартинформ, 2019. — 65 с.

30. **ГОСТ Р ИСО/МЭК 27033-1-2011**. Информационная технология. Методы и средства обеспечения безопасности. Безопасность сетей. Часть 1. Обзор и концепции [Текст]. — Введ. 2012–01–01. — М.: Стандартинформ, 2012. — 73 с.

31. **ГОСТ Р ИСО/МЭК 27033-3-2014**. Информационная технология. Методы и средства обеспечения безопасности. Безопасность сетей. Часть 3. Эталонные се-

тевые сценарии. Угрозы, методы проектирования и вопросы управления [Текст]. — Введ. 2015–11–01. — М.: Стандартинформ, 2019. — 36 с.

32. **ГОСТ Р ИСО/МЭК 27034-1-2014**. Информационная технология. Методы и средства обеспечения безопасности. Безопасность приложений. Часть 1. Обзор и общие понятия [Текст]. — Введ. 2015–06–01. — М.: Стандартинформ, 2015. — 73 с.

33. **ГОСТ Р ИСО/МЭК 29100–2013**. Информационная технология. Методы и средства обеспечения безопасности. Основы обеспечения приватности [Текст]. — Введ. 2015–01–01. — М.: Стандартинформ, 2019. — 23 с.

34. **ГОСТ Р ИСО/МЭК 27001–2006**. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования [Текст]. — Введ. 2006–12–27. — М.: Стандартинформ, 2008. — 26 с.

35. **ГОСТ Р ИСО/МЭК 27002–2012**. Информационная технология. Методы и средства обеспечения безопасности. Свод норм и правил менеджмента информационной безопасности [Текст]. — Введ. 2014–01–01. — М.: Стандартинформ, 2014. — 104 с.

36. **ГОСТ Р ИСО/МЭК 13335-1-2006**. Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий [Текст]. — Введ. 2006–12–19. — М.: Стандартинформ, 2007. — 19 с.

37. **ГОСТ Р ИСО/МЭК 27003–2012**. Информационная технология. Методы и средства обеспечения безопасности. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Руководство по реализации системы менеджмента информационной безопасности [Текст]. — Введ. 2013–12–01. — М.: Стандартинформ, 2014. — 57 с.

38. **ГОСТ Р ИСО/МЭК 27004–2011**. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент информационной безопасности. Измерения [Текст]. — Введ. 2011–12–01. — М.: Стандартинформ, 2012. — 56 с.

39. **ГОСТ Р ИСО/МЭК 27005–2010**. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности [Текст]. — Введ. 2010–11–30. — М.: Стандартинформ, 2011. — 47 с.

40. **ГОСТ Р ИСО/МЭК 27006–2008**. Информационная технология. Методы и средства обеспечения безопасности. Требования к органам, осуществляющим аудит и сертификацию систем менеджмента информационной безопасности [Текст]. — Введ. 2009–10–01. — М.: Стандартинформ, 2019. — 40 с.

41. **ГОСТ Р ИСО/МЭК 27007–2014.** Информационная технология. Методы и средства обеспечения безопасности. Руководства по аудиту систем менеджмента информационной безопасности [Текст]. — Введ. 2015–06–01. — М.: Стандартинформ, 2019. — 27 с.

42. **ГОСТ Р 56045–2014**. Информационная технология. Методы и средства обеспечения безопасности. Рекомендации для аудиторов в отношении мер и средств контроля и управления информационной безопасностью [Текст]. — Введ. 2015–06–01. — М.: Стандартинформ, 2015. — 44 с.

43. **ГОСТ Р ИСО/МЭК 27013–2014**. Информационная технология. Методы и средства обеспечения безопасности. Руководство по совместному использованию стандартов ИСО/МЭК 27001 и ИСО/МЭК 20000–1 [Текст]. — Введ. 2015–09–01. — М.: Стандартинформ, 2020. — 48 с.

44. **ГОСТ Р ИСО/МЭК 27011–2012**. Руководства по менеджменту информационной безопасности для телекоммуникационных организаций на основе ИСО/МЭК 27002 [Текст]. — Введ. 2014–01–01. — М.: Стандартинформ, 2014. — 53 с.

45. **ГОСТ Р ИСО 27799–2015**. Информатизация здоровья. Менеджмент защиты информации в здравоохранении по ИСО/МЭК 27002 [Текст]. — Введ. 2016–11–01. — М.: Стандартинформ, 2016. — 51 с.

46. ФСТЭК России. Методический документ. Профили защиты операционных систем типов «Б» и «В». [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/1328-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-11-maya-2017-g (по состоянию на 28.04.2021).

47. **Методический документ ИТ.ОС.Б4.ПЗ**. Профиль защиты операционных систем типа «Б» четвертого класса защиты [Текст]. — Утв. ФСТЭК России 2017–05–11. — 2017. — 94 с.

48. ФСТЭК России. Методический документ. Профили защиты операционных систем типа «А». [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/1251-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-8-fevralya-2017-g (по состоянию на 28.04.2021).

49. **Методический документ ИТ.ОС.А4.ПЗ**. Профиль защиты операционных систем типа «А» четвертого класса защиты [Текст]. — Утв. ФСТЭК России 2017–02–08. — 2017. — 133 с.

50. ФСТЭК России. Методический документ. Профили защиты межсетевых экранов. [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/1185-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-12-sentyabrya-2016-g (по состоянию на 28.04.2021).

51. **Методический документ ИТ.МЭ.А4.ПЗ**. Профиль защиты межсетевых экранов типа «А» четвертого класса защиты [Текст]. — Утв. ФСТЭК России 2016–09–12. — 2016. — 89 с.

52. ФСТЭК России. Методический документ. Профили защиты средств контроля съемных машинных носителей информации. [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/926-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-1-dekabrya-2014-g (по состоянию на 28.04.2021).

53. **Методический документ ИТ.СКН.Н4.ПЗ**. Профиль защиты средств контроля отчуждения (переноса) информации со съемных машинных носителей информации четвертого класса защиты [Текст]. — Утв. ФСТЭК России 2014–12–01. — 2014. — 50 с.

54. Методический документ. Меры защиты информации в государственных информационных системах [Текст]. — Утв. ФСТЭК России 2014–02–11. — 2014. — 176 с.

55. ФСТЭК России. Методический документ. Профили защиты средств доверенной загрузки. [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/792-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-30-dekabrya-2013-g (по состоянию на 28.04.2021).

56. **Методический документ ИТ.СДЗ.3З5.ПЗ**. Профиль защиты средства доверенной загрузки уровня загрузочной записи пятого класса защиты [Текст]. — Утв. ФСТЭК России 2013–12–30. — 2013. — 45 с.

57. ФСТЭК России. Методический документ. Профили защиты средств антивирусной защиты. [Электронный ресурс] / Электрон. текстовые данные и граф.

данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/470-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-3-fevralya-2012-g-2 (по состоянию на 28.04.2021).

58. **Методический документ ИТ.САВ3.А4.ПЗ**. Профиль защиты средств антивирусной защиты типа «А» четвертого класса защиты [Текст]. — Утв. ФСТЭК России 2012–06–14. — 2012. — 48 с.

59. ФСТЭК России. Методический документ. Профили защиты систем обнаружения вторжений. [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/407-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-6-marta-2012-g (по состоянию на 28.04.2021).

60. **Методический документ ИТ.СОВ.С5.ПЗ**. Профиль защиты систем обнаружения вторжений уровня сети пятого класса защиты [Текст]. — Утв. ФСТЭК России 2012–03–06. — 2012. — 66 с.

61. ФСТЭК России. Методический документ. Профили защиты систем обнаружения вторжений. [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty/406-metodicheskie-dokumenty-utverzhdeny-fstek-rossii-3-fevralya-2012-g (по состоянию на 28.04.2021).

62. **Методический документ ИТ.СОВ.С4.ПЗ**. Профиль защиты систем обнаружения вторжений уровня сети четвертого класса защиты [Текст]. — Утв. ФСТЭК России 2012–02–03. — 2012. — 73 с.

63. Методический документ. Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных (выписка) [Текст]. — Утв. ФСТЭК России 2008–02–15. — 2008. — 69 с.

64. Методический документ. Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных [Текст]. — Утв. ФСТЭК России 2008–02–14. — 2008. — 10 с.

65. Методический документ. Методика оценки угроз безопасности информации [Текст]. — Утв. ФСТЭК России 2021–02–05. — 2021. — 83 с.

66. Гостехкомиссия России. Руководящий документ. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий. [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty/114-spetsialnye-normativnye-dokumenty/381-rukovodyashchij-dokument (по состоянию на 28.04.2021).

67. Гостехкомиссия России. Руководящий документ. Защита от несанкционированного доступа (НСД) к информации Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей. [Текст]. — Утв. решением председателя Государственной технической комиссии при Президенте Российской Федерации от 4 июня 1999 г. N 114. — 1999. — 9 с.

68. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. [Текст]. — Утв. решением председателя Государственной технической комиссии при Президенте Российской Федерации от 25 июля 1997 г. — 1997. — 15 с.

69. Гостехкомиссия России. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации [Текст]. — Утв. решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. — 1992. — 29 с.

70. Гостехкомиссия России. Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения [Текст]. — Утв. решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. — 1992. — 7 с.

71. Гостехкомиссия России. Руководящий документ. Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации [Текст]. — Утв. решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. — 1992. — 7 с.

72. Гостехкомиссия России. Руководящий документ. Руководство по разработке профилей защиты и заданий по безопасности [Текст]. — Гостехкомиссия России. — 2003. — 154 с.

73. Гостехкомиссия России. Руководящий документ. Безопасность информационных технологий. Руководство по формированию семейств профилей защиты [Текст]. — Гостехкомиссия России. — 2003. — 9 с.

74. Гостехкомиссия России. Руководящий документ. Безопасность информационных технологий. Руководство по регистрации профилей защиты [Текст]. — Гостехкомиссия России. — 2003. — 10 с.

75. Гостехкомиссия России. Руководящий документ. Безопасность информационных технологий. Положение по разработке профилей защиты и заданий по безопасности [Текст]. — Гостехкомиссия России. — 2003. — 11 с.

76. Гостехкомиссия России. Руководящий документ. Защита информации. Специальные защитные знаки. Классификация и общие требования [Текст]. — Гостехкомиссия России. — Утв. решением председателя Государственной технической комиссии при Президенте Российской Федерации от 25 июля 1997 г. — 1997. — 4 с.

77. Руководящий документ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа в автоматизированных системах и средствах вычислительной техники [Текст]. — Утв. решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. — 1992. — 21 с.

78. Официальный сайт компании Gartner [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.gartner.com/en (по состоянию на 28.04.2021).

79. Продукт IBM QRadar SIEM. Официальный сайт компании Gartner [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.ibm.com/ru-ru/products/qradar-siem (по состоянию на 28.04.2021).

80. Продукт Splunk Enterprise Security. Официальный сайт компании splunk> [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.splunk.com/en_us/software/enterprise-security.html (по состоянию на 28.04.2021).

81. Продукт Exabeam SIEM. Официальный сайт компании Exabeam [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.exabeam.com/product/ (по состоянию на 28.04.2021).

82. Продукт Securonix SIEM. Официальный сайт компании Securonix [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.securonix.com/ (по состоянию на 28.04.2021).

83. Продукт LogRhythm SIEM. Официальный сайт компании LogRhythm [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://logrhythm.com/products/nextgen-siem-platform/ (по состоянию на 28.04.2021).

84. Продукт InsightIDR. Официальный сайт компании Rapid7 [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.rapid7.com/products/insightidr/ (по состоянию на 28.04.2021).

85. Продукт NetWitness Platform [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.rsa.com/en-us/products/threat-detection-response/siem-security-information-event-management (по состоянию на 28.04.2021).

86. Официальный сайт компании splunk> [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://www.splunk.com/en_us/form/gartner-siem-magic-quadrant.html (по состоянию на 28.04.2021).

87. MASSIF FP7 Project. MAnagement of Security information and events in Service Infrastructures [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — EC FP7–257475. — Режим доступа: https://cordis.europa.eu/project/id/257475 (по состоянию на 28.04.2021).

88. **Котенко И. В.** Интеллектуальные сервисы защиты информации в критических инфраструктурах / И. В. Котенко, И. Б. Саенко, А. А. Чечулин, О. В. Полубелова, Е. С. Новикова, Е. В. Дойникова, А. В. Шоров, В. А. Десницкий; под общей ред. И. В. Котенко, И. Б. Саенко. — СПб.: БХВ-Петербург, 2019. — 400 с. — ISBN 978-5-9775-3968-5.

89. **Kotenko I.** Countermeasure selection in SIEM systems based on the integrated complex of security metrics / I. Kotenko, E. Doynikova // Proceedings of the 23th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2015). — Turku, Finland. 4–6 March, 2015. Los Alamitos, California. — IEEE Computer Society, 2015. — P. 567–574.

90. **Дойникова Е. В.** Динамическое оценивание защищенности компьютерных сетей в SIEM-системах / Е. В. Дойникова, И. В. Котенко, А. А. Чечулин // Безопасность информационных технологий. — 2015. — № 3. — С. 33–42.

91. **Visintine V**. Global information assurance certification paper [Электронный ресурс]. SANS Institute 2003 / V. Visintine. — 13 p. — Режим доступа: http://www.giac.org/paper/gsec/3156/introduction-information-risk-assessment/105258 (по состоянию на 28.04.2021).

92. Проект Security Content Automation Protocol (SCAP). Официальный сайт NIST [Электронный ресурс] / Электрон. текстовые дан. и граф. дан. — [Б. м. : б. и.]. — Режим доступа: https://csrc.nist.gov/projects/security-content-automation-protocol (по состоянию на 28.04.2021).

93. **Waltermire D.** NIST Special Publication 800–126. Revision 3. The Technical Specification for the Security Content Automation Protocol (SCAP). SCAP Version 1.3 [Электронный документ] / D. Waltermire, S. Quinn, H. Booth, K. Scarforne, D. Prisaca. — [Б. м. : б. и.], 2018. — Режим доступа: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-126r3.pdf (по состоянию на 28.04.2021).

94. MITRE Website [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.mitre.org (по состоянию на 28.04.2021).

95. **Grance T.** [Presentation]. Automating Compliance with Security Content Automation Protocol // T. Grance. — NIST. — 2008.

96. **Котенко И. В.** Методы оценивания уязвимостей: использование для анализа защищенности компьютерных систем [Текст] / И. В. Котенко, Е. В. Дойникова // Защита информации. Инсайд. — СПб., 2011 — № 4. — С. 74–81.

97. **Mell P.** A Complete Guide to the Common Vulnerability Scoring System (CVSS) Version 2.0 [Электронный документ] / P. Mell, K. Scarforne, S. Romanosky. — [Б. м. : б. и.], 2007. — Режим доступа: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198 (по состоянию на 28.04.2021).

98. **Котенко И. В.** Система оценки уязвимостей CVSS и ее использование для анализа защищенности компьютерных систем [Текст] / И. В. Котенко, Е. В. Дойникова // Защита информации. Инсайд. — СПб., 2011 — № 5. — С. 54–60.

99. **Doynikova E.** Analytical Attack Modeling and Security Assessment based on the Common Vulnerability Scoring System [Текст] / E. Doynikova, A. Chechulin, I. Kotenko // Proceedings of the 20th Conference of Open Innovations Association FRUCT, LETI University, St. Petersburg, Russia. ISSN 2305–7254, ISBN 978-952-68653-0-0, FRUCT Oy, e-ISSN 2343–0737. — IEEE Xplore, 2017. — P. 53–61.

100. FIRST Org. Inc, Common Vulnerability Scoring System v3.0: Specification Document. 2015 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.first.org/cvss/specification-document (по состоянию на 28.04.2021).

101. FIRST Org. Inc, Common Vulnerability Scoring System v3.1: Specification Document. 2019 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf (по состоянию на 28.04.2021).

102. Microsoft Corporation. Microsoft Security Response Center. Security Update Severity Rating System [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.microsoft.com/en-us/msrc/security-update-severity-rating-system (по состоянию на 28.04.2021).

103. Microsoft Corporation. Microsoft Security Response Center. Microsoft Exploitability Index [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.microsoft.com/en-us/msrc/exploitability-index?rtc=1 (по состоянию на 28.04.2021).

104. nCircle Vulnerability Scoring System [Электронный документ] / nCircle Network Security Inc. — 2009. — 12 p. — Режим доступа: http://index-of.es/z0ro-Repository-3/ncircle_vulnerability_scoring-2.pdf (по состоянию на 28.04.2021).

105. Tripwire white paper. Tripwire Vulnerability Scoring System [Электронный документ] / 3 Tripwire, Inc. — 2013. — 10 p. — Режим доступа: https://dsimg.ubm-us.net/envelope/160343/293772/1396040281_Tripwire_Vulnerability_Scoring_System_white_paper.pdf (по состоянию на 28.04.2021).

106. Common Weakness Enumeration (CWE) [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://cwe.mitre.org/data/index.html (по состоянию на 28.04.2021).

107. MITRE. Schema Documentation — Schema Version 6.4 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — 2021. — Режим доступа: https://cwe.mitre.org/documents/schema/index.html (по состоянию на 28.04.2021).

108. MITRE. Schema Documentation — Schema Version 3.4 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — 2021. — Режим доступа: https://capec.mitre.org/data/xsd/ap_schema_latest.xsd (по состоянию на 27.04.2021).

109. The Web Application Security Consortium. The WASC Threat Classification v2.0 [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — 2021. — Режим доступа: http://projects.webappsec.org/w/page/13246978/Threat%20Classification (по состоянию на 27.04.2021).

110. OWASP. Attacks [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — 2021. — Режим доступа: https://owasp.org/www-community/attacks/ (по состоянию на 27.04.2021).

111. **Barnum S.** Common Attack Pattern Enumeration and Classification (CAPEC) [Текст] / Schema Description. — [Б. м. : б. и.], 2008.

112. MITRE. CAPEC Category: Inject Unexpected Items [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — 2021. — Режим доступа: https://capec.mitre.org/data/definitions/152.html (по состоянию на 27.04.2021).

113. MITRE. CAPEC Category: Traffic Injection [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — 2021. — Режим доступа: https://capec.mitre.org/data/definitions/594.html (по состоянию на 27.04.2021).

114. MITRE. CAPEC-26: Leveraging Race Conditions [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — 2021. — Режим доступа: https://capec.mitre.org/data/definitions/26.html (по состоянию на 27.04.2021).

115. **Pauli J.** Towards a specification prototype for hierarchy-driven attack patterns [Текст] / J. Pauli, P. Engebretson // Information Technology: New Generations. — [Б. м. : б. и.], 2008.

116. **Gamal M. M.** A Security Analysis Framework Powered by an Expert System [Текст] / M. M. Gamal, D. Hasan, A. F. Hegazy // International Journal of Computer Science and Security. — [Б. м. : б. и.], 2011. — Vol. 4, Issue 6. — P. 505–526.

117. MITRE ATT&CK website. Enterprise Matrix [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://attack.mitre.org/matrices/enterprise/ (по состоянию на 27.04.2021).

118. MITRE ATT&CK website. Mobile Matrices [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://attack.mitre.org/matrices/mobile/ (по состоянию на 27.04.2021).

119. MITRE ATT&CK website. Enterprise Mitigations [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://attack.mitre.org/mitigations/enterprise/ (по состоянию на 27.04.2021).

120. MITRE ATT&CK website. Mobile Mitigations [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://attack.mitre.org/mitigations/mobile/ (по состоянию на 27.04.2021).

121. Github website. Набор данных ATT&CK на языке STIX [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://github.com/mitre/cti (по состоянию на 27.04.2021).

122. Github website. Python библиотека для работы с набором данных ATT&CK на языке STIX [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://github.com/oasis-open/cti-python-stix2#installation (по состоянию на 27.04.2021).

123. Github website. MITRE CALDERA [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://github.com/mitre/caldera (по состоянию на 27.04.2021).

124. Github website. MITRE CASCADE [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://github.com/mitre/cascade-server (по состоянию на 27.04.2021).

125. MITRE Cyber Analytics Repository [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://car.mitre.org/ (по состоянию на 27.04.2021).

126. CAPEC Website. ATT&CK Comparison [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://capec.mitre.org/about/attack_comparison.html (по состоянию на 27.04.2021).

127. **Gerard T. M.** Common Remediation Enumeration (CRE) Version 1.0 (Draft). NIST Interagency Report 7831 (Draft) / T. M. Gerard, D. Waltermire, J. O. Baker ; National Institute of Standards and Technology, U. S. Department of Commerce. — [Б. м. : б. и.], 2011.

128. **Johnson C.** Enterprise Remediation Automation [Текст] / C. Johnson ; NIST // Proceedings of the IT Security Automation Conference (September 27–29, 2010).

129. NIST website. Computer Security Resource Center. Control Catalog spreadsheet [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final (по состоянию на 27.04.2021).

130. **Singhal A.** Security risk analysis of enterprise networks using probabilistic attack graphs. NIST Interagency Report 7788 / A. Singhal, X. Ou. — Gaithersburg: National Institute of Standards and Technology, 2011. — 24 p.

131. The Center for Internet Security. The CIS Security Metrics [Текст]. — The Center for Internet Security, 2009. — 175 p.

132. **Dacier M.** Quantitative Assessment of Operational Security-Models and Tools [Текст]. LAAS Research Report 96493 / M. Dacier, Y. Deswarte et al. — 1996.

133. **Liu, Y.** Network vulnerability assessment using Bayesian networks / Y. Liu, H. Man // Proceedings of the SPIE. — Vol. 5812. — 2005. — P. 61–71.

134. **Man D.** A quantitative evaluation model for network security [Текст] / D. Man, W. Yang, Y. Yang, W. Wang, L. Zhang // Proceedings of the 2007 International Conference on Computational Intelligence and Security (December 15–19, 2007). — P. 773–777.

135. **Wu Y.-S.** Automated adaptive intrusion containment in systems of interacting services / Y.-S. Wu, B. Foo, Y.-C. Mao, S. Bagchi, E. H. Spafford // Computer Networks: The International Journal of Computer and Telecommunications Networking. — 2007. — Vol. 51. — P. 1334–1360.

136. **Stakhanova N.** A cost-sensitive model for preemptive intrusion response systems / N. Stakhanova, S. Basu, J. Wong // Proceedings of the 21st International Conference on Advanced Networking and Applications. — 2007.

137. **Frigault M.** Measuring network security using dynamic Bayesian network / M. Frigault, L. Wang, A. Singhal, S. Jajodia // Proceedings of the ACM Workshop on Quality of Protection (October 2008). — [Б. м. : б. и.], 2008.

138. **Dantu R.** Network risk management using attacker profiling [Текст] / R. Dantu, P. Kolan, J. Cangussu // Security and Communication Networks. — [Б. м. : б. и.], 2009. — Vol. 2, No. 1. — P. 83–96.

139. **Wang L.** An Attack Graph-Based Probabilistic Security Metric / L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia // Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security. — Heidelberg: Springer-Verlag Berlin, 2008. — P. 283–296. — DOI: 10.1007/978-3-540-70567-3_22.

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

141. **Frigault M.** Measuring the Overall Network Security by Combining CVSS Scores Based on Attack Graphs and Bayesian Networks / M. Frigault, L. Wang, S. Jajodia, A. Singhal // In book: Network Security Metrics. — 2017. — P. 1–23.

142. **Muñoz-González L.** Exact Inference Techniques for the Analysis of Bayesian Attack Graphs / L. Muñoz-González, D. Sgandurra, M. Barrère, E. C. Lupu // IEEE Transactions on Dependable and Secure Computing. — 16. — 2019. — P. 231–244.

143. **Khosravi-Farmad M.** Bayesian Decision Network-Based Security Risk Management Framework / M. Khosravi-Farmad, A. Ghaemi-Bafghi // Journal of Network and Systems Management, 2020. — P. 1–26.

144. **Toth T.** Evaluating the impact of automated intrusion response mechanisms / T. Toth, C. Kruegel // Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC). — 2002. — P. 301–310.

145. **Balepin I.** Using specification-based intrusion detection for automated response [Текст] / I. Balepin, S. Maltsev, J. Rowe, K. Levitt // Proceedings of the

sixth International Symposium on Recent Advances in Intrusion Detection (RAID). — [Б. м. : б. и.], 2003. — P. 136–154.

146. **Kheir N.** Cost evaluation for intrusion response using dependency graphs [Текст] / N. Kheir, H. Debar, N. Cuppens-Boulahia, F. Cuppens, J. Viinikka / Proceedings of the International Conference on Network and Service Security (Paris, June 24–26, 2009). — IEEE, 2009. — P. 1–6.

147. **Jahnke M.** Graph-based metrics for intrusion response measures in computer networks / M. Jahnke, C. Thul, P. Martini // Proceedings of the IEEE Workshop on Network Security (2007).

148. **Kheir N.** A service dependency model for cost-sensitive intrusion response / N. Kheir, N. Cuppens-Boulahia, F. Cuppens, H. Debar // Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS'10). — Vol. 6345. — 2010. — P. 626–642.

149. **Chunlu W.** A novel comprehensive network security assessment approach [Текст] / W. Chunlu, W. Yancheng, D. Yingfei Z. Tianle // Proceedings of the 2011 IEEE International Conference on Communications (Kyoto). — IEEE, 2011. — P. 1–6.

150. **Ahmed M. S.** A novel quantitative approach for measuring network security [Текст] / M. S. Ahmed, E. Al-Shaer, L. Khan. — Proceedings of the INFOCOM'08. — [Б. м. : б. и.], 2008. — P. 1957–1965.

151. **He W.** Unknown Vulnerability Risk Assessment Based on Directed Graph Models: A Survey [Текст] / He, W., Li, H., J. Li // IEEE Access. — 7. — IEEE, 2019.

152. **Vaughn R.** Information assurance measures and metrics: State of Practice and Proposed Taxonomy / R. Vaughn, R. Henning, A. Siraj // Proceedings of the 36th Hawaii Int. Conf. on System Sciences (HICSS 03). — 2003.

153. **Henning R.** Security Metrics / R. Henning, et al. ; MITRE // Proceedings of the Workshop on Information Security System, Scoring and Ranking (Williamsburg, Virginia). — 2002.

154. **Seddigh N.** Current trends and advances in information assurance metrics / N. Seddigh, P. Pieda, A. Matrawy, B. Nandy, I. Lambadaris, A. Hatfield // Proceedings of the 2nd Annual Conference on Privacy, Security and Trust (Fredericton, NB, October 2004). — 2004.

155. **Chew E.** Performance Measurement Guide for Information Security. NIST Special Publication 800–55 / E. Chew, M. M. Swanson, K. M. Stine, N. Bartol, A. Brown, W. Robinson. — 2008. — Режим доступа: https://www.nist.gov/publications/performance-measurement-guide-information-security?pub_id=152183 (по состоянию на 27.04.2021).

156. **Idika N. C.** Characterizing and aggregating attack graph-based security metric : PhD Thesis / N. C. Idika. — Purdue University, 2010. — 131 p.

157. **Axelrod C. W.** Accounting for value and uncertainty in security metrics [Текст] / C. W Axelrod // Information Systems Control Journal. — [Б. м. : б. и.], 2008. — Vol. 6. — P. 1–6.

158. **Kanoun W.** Automated reaction based on risk analysis and attackers skills in intrusion detection systems [Текст] / W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, J. Araujo // Proceedings of the Third International Conference on Risks and Security of Internet and Systems (October 28–30, 2008). — P. 117–124.

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

160. **Caralli R. A.** Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process [Текст]. Technical Report / R. A. Caralli, J. F. Stevens, L. R. Young, W. R. Wilson. — Software Engineering Institute, 2007. — P. 154.

161. OCTAVE [Электронный ресурс] / CERT website. — Электрон. текстовые данные и граф. данные. — Режим доступа: https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=8419 (по состоянию на 27.04.2021).

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

163. **Artz M.** NetSPA, a network security planning architecture [Текст] : Master's thesis / M. Artz. — Massachusetts Institute of Technology, 2002.

164. **Lippmann R. P.** Validating and restoring defense in depth using attack graphs [Текст] / R. P. Lippmann et al // Proceedings of MILCOM 2006 (Washington, DC).

165. **Ingols K.** Practical attack graph generation for network defense [Текст] / K. Ingols, R. Lippmann, K. Piwowarski // Proceedings of 22nd Annual Conference on the Computer Security Applications (Miami Beach, FL, 2006). — IEEE, 2006. — P. 121–130.

166. **Чечулин А. А.** Построение и анализ деревьев атак на компьютерные сети с учетом требования оперативности [Текст]: диссертация кандидата технических наук: 05.13.19 / Чечулин Андрей Алексеевич [Место защиты: С.-Петерб. ин-т информатики и автоматизации РАН]. — СПб., 2013. — 152 с.: ил. Методы и системы защиты информации, информационная безопасность. Хранение: 61 14-5/933.

167. **Котенко И. В.** Метрики безопасности для оценки уровня защищенности компьютерных сетей [Текст] / И. В. Котенко, М. В. Степашкин // Защита информации. Инсайд. — СПб., 2006. — № 3.

168. **Ou X.** MULVAL: A logic based network security analyzer / X. Ou, S. Govindavajhala, A. W. Apple // Proceedings of the 14th USENIX Security Symposium. — 2005.

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

170. **Portier B.** SOA terminology overview, Part 1: Service, architecture, governance, and business terms / B. Portier. — 2008.

171. Web Services Glossary. W3C Working Group Note 11 February 2004 [Электронный ресурс] / eds.: H. Haas, A. Brown. — Режим доступа: http://www.w3.org/TR/ws-gloss (по состоянию на 27.04.2021).

172. RiskWatch [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.riskwatch.com (по состоянию на 27.04.2021).

173. Github website. Microsoft. Attack Surface Analyzer [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://github.com/Microsoft/AttackSurfaceAnalyzer (по состоянию на 27.04.2021).

174. **Howard M.** Measuring relative attack surfaces [Электронный ресурс] / M. Howard, J. Pincus, J. M. Wing // Proceedings of Workshop on Advanced Developments in Software and Systems Security (Taipei, 2003). — Режим доступа: http://www.cs.cmu.edu/~wing/publications/Howard-Wing03.pdf (по состоянию на 27.04.2021).

175. **Manadhata P. K.** Measuring a system's attack surface [Электронный ресурс] / P. K. Manadhata. — PA: Carnegie Mellon University, 2004. — Режим доступа: http://reports-archive.adm.cs.cmu.edu/anon/2004/CMU-CS-04–102.pdf (по состоянию на 27.04.2021).

176. **Manadhata P. K.** A formal model for a system's attack surface [Электронный ресурс] / P. K. Manadhata, D. K. Kaynar, J. M. Wing. — Pittsburgh, PA: Carnegie Mellon University, 2007. — Режим доступа: http://www.cs.cmu.edu/~wing/publications/CMU-CS-07–144.pdf (по состоянию на 27.04.2021).

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

178. **Yazar Z.** A Qualitative Risk Analysis and Management Tool — Cramm / Z. Yazar // SANS Institute Information Security Reading Room, 2021. — Режим доступа: https://www.sans.org/reading-room/whitepapers/auditing/qualitative-risk-analysis-management-tool-cramm-83 (по состоянию на 27.04.2021).

179. **Noel S.** Understanding Complex Network Attack Graphs through Clustered Adjacency Matrices / S. Noel, J. Jajodia // in Proceedings of the 21st Annual Computer Security Applications Conference. — 2005.

180. **Noel S.** Managing Attack Graph Complexity through Visual Hierarchical Aggregation / S. Noel, S. Jajodia // in Proceedings of the ACM CCS Workshop on Visualization and Data Mining for Computer Security. — 2004.

181. **Jajodia S.** Topological Analysis of Network Attack Vulnerability / S. Jajodia, S. Noel, B. O'Berry // in Managing Cyber Threats: Issues, Approaches and Challenges, V. Kumar, J. Srivastava, A. Lazarevic (eds.). — Springer, 2005.

182. **Ingols K.** Modeling Modern Network Attacks and Countermeasures Using Attack Graphs / K. Ingols, M. Chu, R. Lippmann, S. Webster and S. Boyer // Proceedings of ACSAC Conference. — 2009.

183. **Ou X.** A Scalable Approach to Attack Graph Generation / X. Ou, W. F. Boyer, M. A. McQueen // Proceedings of 13th ACM CCS Conference. — 2006. — P. 336–345.

184. Amenaza website. SecurITree [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.amenaza.com/ (по состоянию на 27.04.2021).

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

186. **He W.** A network security risk assessment framework based on game theory / W. He, C. Xia, C. Zhang, Y. Ji, X. Ma // Proceedings of the Second International Conference on Future Generation Communication and Networking, FGCN '08 (Hainan Island, December 13–15, 2008). — Vol. 2. — IEEE, 2009. — P. 249–253.

187. **Strasburg C.** Intrusion response cost assessment methodology / C. Strasburg, N. Stakhanova, S. Basu, J. S. Wong // Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (NY, USA, 2009). — 2009. — P. 388–391.

188. **Bursztein E.** Using strategy objectives for network security analysis [Текст] / E. Bursztein, J. C. Mitchell // Information Security and Cryptology ; series: Lecture Notes in Computer Science. — Vol. 6151. — Springer Berlin Heidelberg, 2010. — P. 337–349.

189. **Granadillo G. G.** Individual countermeasure selection based on the return on response investment index / G. G. Granadillo, H. Debar, G. Jacob, M. Achemlal // LNCS, Computer Network Security, proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012 (St. Petersburg, Russia, October 17–19, 2012). — Vol. 7531. — Springer, 2012. — P. 156–170. — DOI: 10.1007/978-3-642-33704-8_14.

190. **Kotenko I.** Attack graph based evaluation of network security / I. Kotenko, M. Stepashkin // Proceedings of the 10th IFIP Conference on Communications and Multimedia Security (Heraklion, Greece, 2006). — 2006. — P. 216–227.

191. **Hoo K. J. S.** How Much is Enough? A Risk-Management Approach to Computer Security [Текст]: PhD thesis / K. J. S. Hoo. — Stanford University, 2000.

192. **Cremonini M.** Evaluating information security investments from attackers perspective: the Return-On-Attack (ROA) [Текст] / M. Cremonini, P. Martini // Proceedings of the Fourth Workshop on the Economics of Information Security (June 2–3, 2005). — [Б. м. : б. и.], 2005.

193. CyVision Website. Система Cauldron [Электронный ресурс]. — Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.benvenisti.net/ (по состоянию на 27.04.2021).

194. **Jajodia S.** Review of CAULDRON Tool / Sushil Jajodia, Steven Noel // NSA, Linthicum, Maryland, March 2009.

195. **Дойникова Е. В.** Оценка защищенности и выбор защитных мер в компьютерных сетях на основе графов атак и зависимостей сервисов [Текст]: диссертация кандидата технических наук: 05.13.19 / Дойникова, Елена Владимировна; [Место защиты: С.-Петерб.]. — Санкт-Петербург, 2017. — Методы и системы защиты информации, информационная безопасность.

196. **Дойникова Е. В.** Показатели и методики оценки защищенности компьютерных сетей на основе графов атак и графов зависимостей сервисов / Е. В. Дойникова // Труды СПИИРАН. — Вып. 3 (26). — СПб.: Наука, 2013. — С. 54–68.

197. Information Security Metrics. State of the Art [Текст] / DSV Report; writer: R. Barabanov, eds.: S. Kowalski, L. Yngström. — No. 11–007. — [Б. м. : б. и.], 2011.

198. **McIntyre A.** I3P Research report No. 12. Security metrics tools final report [Электронный документ] / A. McIntyre, B. Becker, D. Bodeau, B. Gennert, C. Glantz, L. R. O'Neil, J. R. Santos, M. Stoddard // Institute for Information Infrastructure Protection. — [Б. м. : б. и.], 2007. — Режим доступа: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.170.1610&rep=rep1&type=pdf (по состоянию на 27.04.2021).

199. **Swanson M.** Security metrics guide for information technology systems. NIST Special Publication 800–55 / M. Swanson, N. Bartol, J. Sabato, J. Hash, L. Graffo. — 2003.

200. **Doynikova E.** CVSS-based Probabilistic Risk Assessment for Cyber Situational Awareness and Countermeasure Selection / E. Doynikova, I. Kotenko // Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2017). — St. Petersburg, Russia, March 6–8, 2017. Los Alamitos, California. — IEEE Computer Society, 2017. — P. 346–353.

201. **Doynikova E.** The multi-layer graph based technique for proactive automatic response against cyber attacks / E. Doynikova, I. Kotenko // Proceedings of the 26th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2018). Cambridge, UK, March 21–23, 2018. Los Alamitos, California. — IEEE Computer Society, 2018. — P. 470–477. — DOI: 10.1109/PDP2018.2018.00081.

202. **Kotenko I.** Selection of countermeasures against network attacks based on dynamical calculation of security metrics / I. Kotenko, E. Doynikova // Journal of Defense Modeling and Simulation. — Vol. 15, Issue 2. — 2018. — P. 181–204. — DOI: 10.1177/1548512917690278.

203. A Guide for Government Agencies Calculating Return on Security Investment. Version 2.0 [Текст] / Government Chief Information Office (GCIO). — Lockstep Consulting, 2014. — 33 p.

204. **Doynikova E.** Countermeasure selection based on the attack and service dependency graphs for security incident management / E. Doynikova, I. Kotenko // 10th International Conference on Risks and Security of Internet and Systems : CRiSIS 2015. July 20–22, Mytilene, Lesvos Island, Greece / C. Lambrinoudakis and A. Gabillon (Eds.). Lecture Notes in Computer Science (LNCS). — Vol. 9572. — Springer, 2016. — P. 107–124. — DOI: 10.1007/978-3-319-31811-0_7. http://link.springer.com/chapter/10.1007/978-3-319-31811-0_7.

205. NMap reference guide [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://nmap.org/book/man.html (по состоянию на 27.04.2021).

206. Nessus vulnerability scanner [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.tenable.com/products/nessus-vulnerability-scanner (по состоянию на 27.04.2021).

207. Wireshark vulnerability scanner [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.wireshark.org (по состоянию на 27.04.2021).

208. **Дойникова Е. В.** Совершенствование графов атак для мониторинга кибербезопасности: оперирование неточностями, обработка циклов, отображение инцидентов и автоматический выбор защитных мер / Е. В. Дойникова, И. В. Котенко // Труды СПИИРАН, 2018. — Вып. 2(57). — С. 211–240. — DOI: 10.15622/sp.57.9.

209. **Степашкин М. В.** Модели и методика анализа защищенности компьютерных сетей на основе построения деревьев атак [Текст]: диссертация кандидата технических наук: 05.13.11, 05.13.19 / Степашкин, Михаил Викторович; [Место защиты: С.-Петерб.]. — Санкт-Петербург, 2007. — 196 с.: ил. Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

210. **Kheir N.** Response policies & counter-measures: Management of service dependencies and intrusion and reaction impacts : PhD Thesis / N. Kheir. — Telecom Bretagne, 2010.

211. **Howard J.** A Common Language for Computer Security Incidents. SANDIA Report / J. Howard, T. Longstaff. — SAND98–8667. — 1998.

212. FreeNATS [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.purplepixie.org/freenats (по состоянию на 27.04.2021).

213. NetCrunch [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.adremsoft.com/netcrunch (по состоянию на 27.04.2021).

214. Intrust [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.quest.com/intrust (по состоянию на 27.04.2021).

215. Snort [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://www.snort.org (по состоянию на 27.04.2021).

216. Cisco Secure Intrusion Detection System [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/prod_eol_notice09186a008009230e.html (по состоянию на 27.04.2021).

217. Kaspersky [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://usa.kaspersky.com/products-services/home-computer-security/anti-virus (по состоянию на 27.04.2021).

218. Comodo [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.comodo.com (по состоянию на 27.04.2021).

219. Endian [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: http://www.endian.com (по состоянию на 27.04.2021).

220. X-Force [Электронный ресурс] / Электрон. текстовые данные и граф. данные. — Режим доступа: https://exchange.xforce.ibmcloud.com (по состоянию на 27.04.2021).

221. **Новикова Е. С.** Проектирование компонента визуализации для автоматизированной системы управления информационной безопасностью [Текст] / Е. С. Новикова, И. В. Котенко // Информационные технологии. — Новые технологии, 2013. — № 9. — С. 32–36.

Приложение

Приложение А

Таблица А.1

Перечисления, используемых в схеме CWE [106]

Название перечисления	Возможные значения	Описание значений
Вероятности (LikelihoodEnumeration)	Высокая (High)	
	Средняя (Medium)	
	Низкая (Low)	
	Неизвестная (Unknown)	
Уровни абстракции (AbstractionEnumeration)	Столп (Pillar)	Наивысший уровень абстракции, отображает тему всех слабых мест с уровнем абстракции «класс» (class), «основа» (base), «вариант» (variant).
	Класс (Class)	Слабое место описывается абстрактно (абстрактнее, чем в случае абстракции Base), независимо от языка и технологии, но более конкретно, чем в случае уровня абстракции Pillar. Обычно для описания слабого места используется 1 или 2 из следующих аспектов: поведение, свойство, ресурс.
	Основа (Base)	Более конкретный тип слабого места, в большинстве случаев независимый от языка и технологии, но с достаточной детализацией, позволяющей определить методы обнаружения и предотвращения применения. Обычно для описания слабого места используется от 2 до 3 из следующих аспектов: поведение, свойство, технология, язык, ресурс.
	Вариант (Variant)	Слабое место связано с определенным типом продукта, конкретным языком и технологией. Описывается более конкретно, чем в случае уровня абстракции Base. Обычно для описания слабого места используется от 3 до 5 из следующих аспектов: поведение, свойство, технология, язык, ресурс.
	Составной (Compound)	Значимое объединение нескольких слабых мест.
Структуры (StructureEnumeration)	Цепочка (Chain)	Набор слабых мест, которые должны быть доступны последовательно, чтобы возникла эксплуатируемая уязвимость.
	Составной (Composite)	Набор слабых мест, которые должны одновременно присутствовать, чтобы возникла эксплуатируемая уязвимость.
	Простой (Simple)	Отдельное слабое место, использование которого не зависит от присутствия другого слабого места.

Название перечисления	Возможные значения	Описание значений
Статусы (StatusEnumeration)	Устарела (Deprecated)	Запись, удаленная из CWE, т.к. является дубликатом или была создана по ошибке.
	Черновик (Draft)	Все важные элементы записи заполнены, и критичные элементы, такие как *имя* и *описание*, хорошо описаны.
	Не завершена (Incomplete)	Не все важные элементы записи заполнены, и нет гарантии качества.
	Вышла из употребления (Obsolete)	Запись корректна, но не является актуальной, например, ввиду замены записью новее.
	Стабильная (Stable)	Все важные элементы верифицированы, и запись, скорее всего, не изменится в будущем.
	Употребимая (Usable)	Проведена полноценная проверка, и верифицированы критичные элементы.
Представления (ViewTypeEnumeration)	Неявное (Implicit)	Представляет собой срез, включающий слабые места, связанные конкретным атрибутом, указанным в элементе представления *фильтр* (Filter), например, все слабые места со статусом Draft.
	Явное (Explicit)	Представляет собой срез, включающий слабые места, выделенные в соответствии с некоторым внешним фактором, например, все слабые места, отображаемые на одну из внешних таксономий.
	Граф (Graph)	Иерархическое отображение слабых мест, основанное на конкретной точке зрения, которую может занять пользователь. Зачастую иерархия начинается с категории, за которой следует слабое место уровня абстракции «класс» или «основа», и заканчивается слабым местом уровня абстракции «вариант».
Названия языка (LanguageNameEnumeration)	Ada	
	ASP	
	ASP.NET	
	Basic	
	C	
	C++	
	C#	
	COBOL	
	Fortran	
	F#	
	Go	
	HTML	

Название перечисления	Возможные значения	Описание значений
Названия языка (LanguageNameEnumeration)	Java	
	JavaScript	
	JSP	
	Objective-C	
	Pascal	
	Perl	
	PHP	
	Python	
	Ruby	
	Shell	
	SQL	
	Swift	
	VB.NET	
	Verilog	
	VHDL	
	XML	
	Другой (Other)	
Классы языка (LanguageClassEnumeration)	Ассемблер (Assembly)	
	Компилируемый (Compiled)	
	Интерпретируемый (Interpreted)	
	Независимый от языка (Language-Independent)	Используется для всех языков.
Виды распространенности (PrevalenceEnumeration)	Часто (Often)	
	Иногда (Sometimes)	
	Редко (Rarely)	
	Не определено (Undetermined)	
Названия ОС (OperatingSystem NameEnumeration)	AIX	
	Android	
	BlackBerry OS	
	Chrome OS	
	Darwin	
	FreeBSD	
	iOS	
	macOS	
	NetBSD	
	OpenBSD	
	Red Hat	
	Solaris	
	SUSE	
	tvOS	
	Ubuntu	

Название перечисления	Возможные значения	Описание значений
Названия ОС (OperatingSystem NameEnumeration)	watchOS	
	Windows 9x	
	Windows Embedded	
	Windows NT	
Классы ОС (OperatingSystemCla ssEnumeration)	Linux	
	macOS	
	Unix	
	Windows	
	Не зависит от ОС (OS-Independent)	Используется для всех ОС.
Названия архитектур (ArchitectureNameEnumeration)	Alpha	
	ARM	
	Itanium	
	Power Architecture	
	SPARC	
	x86	
	Другое (Other)	
Классы архитектур (ArchitectureClassEnumeration)	Встраиваемая (Embedded)	
	Микрокомпьютер (Microcomputer)	
	Рабочая станция (Workstation)	
	Не зависит от архитектуры (Architecture-Independent)	Используется для всех архитектур.
Названия технологий (TechnologyNameEnumeration)	Веб-сервер (Web Server)	
	Сервер баз данных (Database Server)	
	Катализирующая интеллектуальная собственность (Accelerator IP (Intellectual Property))	Интеллектуальная собственность, предназначенная для избавления от конкретной рабочей нагрузки для улучшения производительности: DSP, обработка пакетов, математика, сжатие и т. п.
	Интеллектуальная собственность на аналоговые и смешанные сигналы (Analog and Mixed Signal IP)	Интеллектуальная собственность, которая контролирует/определяет электрические сети для связи, которая принимает/передает сигналы, кондиционированные вне цифрового домена системы.
	Интеллектуальная собственность аудио/видео (Audio/Video IP)	Интеллектуальная собственность, разработанная для работы с аудио/видеоданными: кодеры/декодеры, распознавание речи, преобразователи форматов и т. п.
	Интеллектуальная собственность шины/интерфейсов (Bus/Interface IP)	Интеллектуальная собственность, реализующая взаимосвязь между элементами вычислительной системы: I2C, PCIe, DDR, MMC, USB, GPIO, NoC и т. п.

Название перечисления	Возможные значения	Описание значений
Названия технологий (TechnologyNameEnumeration)	Интеллектуальная собственность часов/ счетчика (Clock/Counter IP)	Интеллектуальная собственность, отражающая течение времени в колебаниях или человеческих единицах: часы реального времени, сторожевой таймер, монотонный счетчик и т. п.
	Интеллектуальная собственность коммуникации (Communication IP)	Интеллектуальная собственность, разработанная для передачи/ получения информации: модулятор/ демодулятор, GPS, 802.11, Bluetooth, CDMA/DSM и т. п.
	Интеллектуальная собственность контроллера (Controller IP)	Интеллектуальная собственность жестко запрограммированной схемы (например, конечный автомат) для реагирования в системе управления с обратной связью или в другом ограниченном контексте для управления другим объектом: Arbiter, улучшенный программируемый контроллер прерываний (APIC), USB, периферийное устройство, память, хранилище и т. п.
	Интеллектуальная собственность памяти (Memory IP)	Интеллектуальная собственность, реализующая энергозависимое (временное) хранилище данных: DRAM, SRAM и т. п.
	Интеллектуальная собственность микроконтроллера (Microcontroller IP)	Интеллектуальная собственность, реализующая специализированный процессор, действующий как программируемый контроллер.
	Интеллектуальная собственность сети на кристалле (Network on Chip IP)	
	Интеллектуальная собственность управления питанием (Power Management IP)	Интеллектуальная собственность, которая контролирует и/ или отслеживает состояние питания системы: регуляторы напряжения, контроллеры питания, мониторы питания и т. д.
	Интеллектуальная собственность процессора (Processor IP)	Интеллектуальная собственность, реализующая вычислительную систему общего назначения: CPU, GPU, RISC, CISC и т. п.
	Интеллектуальная собственность безопасности (Security IP)	Интеллектуальная собственность, разработанная для защиты активов: криптография, аутентификация, обнаружение несанкционированного доступа и т. п.
	Интеллектуальная собственность сенсора (Sensor IP)	
	Интеллектуальная собственность хранилища (Storage IP)	

Название перечисления	Возможные значения	Описание значений
Названия технологий (TechnologyNameEnumeration)	Интеллектуальная собственность тестирования/отладки (Test/Debug IP)	Интеллектуальная собственность, разработанная для верификации функциональности и определения причин дефектов: JTAG, BIST, периферийное сканирование, генератор шаблонов и т. п.
	Другое (Other)	
Классы технологий (TechnologyClassEnumeration)	Клиент-сервер (Client Server)	Распределенное приложение, но для целей CWE не используется веб-браузер.
	Облачные вычисления (Cloud Computing)	Хранилище данных и вычислительные мощности предоставляются множеству пользователей через Интернет.
	Мэйнфрейм (Mainframe)	
	Мобильные технологии (Mobile)	
	Многоуровневая архитектура (N-Tier)	
	Сервис-ориентированная архитектура (SOA)	
	Система на чипе (System on Chip)	Технология, которая объединяет все компоненты компьютера на одной интегральной схеме, включая программируемые логически интегральные схемы (FPGA) и интегральные схемы специального назначения (ASIC).
	Веб (Web Based)	Приложения или одностраничные сайты, которые используют веб-браузер для взаимодействия с клиентом.
	Не зависит от технологии (Technology-Independent)	Используется для всех технологий.
Ресурсы (ResourceEnumeration)	CPU	
	Файл или каталог (File or Directory)	
	Память (Memory)	
	Системный процесс (System Process)	
	Другое (Other)	
Области действия (ScopeEnumeration)	Конфиденциальность (Confidentiality)	
	Целостность (Integrity)	
	Доступность (Availability)	
	Контроль доступа (Access Control)	
	Подотчетность (Accountability)	
	Аутентификация (Authentication)	

Название перечисления	Возможные значения	Описание значений
Области действия (ScopeEnumeration)	Авторизация (Authorization)	
	Неотказуемость (Non-Repudiation)	
	Другое (Other)	
Технический ущерб (TechnicalImpactEnumeration)	Изменение памяти (Modify Memory)	
	Чтение памяти (Read Memory)	
	Изменение файлов или диалогов (Modify Files or Directories)	
	Чтение файлов или каталогов (Read Files or Directories)	
	Изменение данных приложений (Modify Application Data)	
	Чтение данных приложений (Read Application Data)	
	Отказ в обслуживании: сбой, выход или перезапуск (DoS: Crash, Exit, or Restart)	
	Отказ в обслуживании: усиление (DoS: Amplification)	
	Отказ в обслуживании: нестабильность (DoS: Instability)	
	Отказ в обслуживании: истощение ресурсов (CPU) (DoS: Resource Consumption (CPU))	
	Отказ в обслуживании: истощение ресурсов (память) (DoS: Resource Consumption (Memory))	
	Отказ в обслуживании: истощение ресурсов (другое) (DoS: Resource Consumption (Other))	
	Запуск неавторизованного кода или команд (Execute Unauthorized Code or Commands)	
	Получение привилегий или присвоение личности (Gain Privileges or Assume Identity)	
	Обход механизма защиты (Bypass Protection Mechanism)	

Название перечисления	Возможные значения	Описание значений
Технический ущерб (TechnicalImpactEnumeration)	Скрытие активности (Hide Activities)	
	Обход логики выполнения (Alter Execution Logic)	
	Ухудшение качества (Quality Degradation)	
Технический ущерб (TechnicalImpactEnumeration)	Неожиданное состояние (Unexpected State)	
	Ущерб зависит от контекста (Varies by Context)	
	Снижение поддерживаемости (Reduce Maintainability)	
	Снижение производительности (Reduce Performance)	
	Снижение надежности (Reduce Reliability)	
	Другое (Other)	
Важность (ImportanceEnumeration)	Нормальная (Normal)	
	Критичная (Critical)	
Методы обнаружения (DetectionMethodEnumeration)	Автоматический анализ (Automated Analysis)	
	Автоматический динамический анализ (Automated Dynamic Analysis)	
	Автоматический статический анализ (Automated Static Analysis)	
	Автоматический статический анализ — исходный код (Automated Static Analysis — Source Code)	
	Автоматический статический анализ — бинарный или байт-код (Automated Static Analysis — Binary or Bytecode)	
	Фаззинг (Fuzzing)	
	Ручной анализ (Manual Analysis)	
	Ручной динамический анализ (Manual Dynamic Analysis)	
	Ручной статический анализ (Manual Static Analysis)	

Название перечисления	Возможные значения	Описание значений
Методы обнаружения (DetectionMethodEnumeration)	Ручной статический анализ — исходный код (Manual Static Analysis — Source Code)	
	Ручной статический анализ — бинарный или байт-код (Manual Static Analysis — Binary or Bytecode)	
	«Белый ящик» (White Box)	
	«Черный ящик» (Black Box)	
Методы обнаружения (DetectionMethodEnumeration)	Обзор архитектуры или дизайна (Architecture or Design Review)	
	Динамический анализ с ручной интерпретацией результатов (Dynamic Analysis with Manual Results Interpretation)	
	Динамический анализ с автоматической интерпретацией результатов (Dynamic Analysis with Automated Results Interpretation)	
	Другое (Other)	
Эффективность (DetectionEffectivenessEnumeration)	Высокая (High)	Метод обнаружения слабого места часто успешен и не приводит к большому количеству ложных отчетов.
	Средняя (Moderate)	Метод применим при множестве обстоятельств, но может не покрывать полностью слабое место или приводит к большому количеству некорректных отчетов.
	Частичная по SOAR (SOAR Partial)	Метод может быть эффективным по стоимости для частичного покрытия цели согласно SOAR (Security Orchestration, Automation and Response).
	Оппортунистическая (Opportunistic)	Метод не направлен напрямую на слабое место, но может быть успешным (случайно, а не надежно).
	Ограниченная (Limited)	Метод может быть полезен при ограниченных обстоятельствах, применим только к подмножеству потенциальных слабых мест, требует обучения/настройки или дает ограниченную видимость.
	Нет (None)	Метод, который, скорее всего, не сработает.

Название перечисления	Возможные значения	Описание значений
Функциональные области (FunctionalAreaEnumeration)	Аутентификация (Authentication)	
	Авторизация (Authorization)	
	Библиотеки кода (Code Libraries)	
	Счетчики (Counters)	
	Криптография (Cryptography)	
	Обработка ошибок (Error Handling)	
	Взаимодействие между процессами (Interprocess Communication)	
	Обработка файлов (File Processing)	
	Журналирование (Logging)	
	Управление памятью (Memory Management)	
	Сети (Networking)	
	Обработка чисел (Number Processing)	
	Вызов программ (Program Invocation)	
	Механизм защиты (Protection Mechanism)	
	Управление сессиями (Session Management)	
	Сигналы (Signals)	
	Обработка строк (String Processing)	
	Не зависит от функциональной области (Functional-Area-Independent)	Используется для всех функциональных областей.
	Питание (Power)	
	Часы (Clock)	
Фазы (PhaseEnumeration)	Политика (Policy)	
	Требования (Requirements)	
	Архитектура и проектирование (Architecture and Design)	
	Реализация (Implementation)	
	Сборка и компиляция (Build and Compilation)	
	Тестирование (Testing)	
	Документирование (Documentation)	
	Комплектация (Bundling)	

Название перечисления	Возможные значения	Описание значений
Фазы (PhaseEnumeration)	Распространение (Distribution)	
	Установка (Installation)	
	Конфигурирование системы (System Configuration)	
	Эксплуатация (Operation)	
	Исправления и обслуживание (Patching and Maintenance)	
	Портирование (Porting)	
	Интеграция (Integration)	
	Производство (Manufacturing)	
Типы заметок (NoteTypeEnumeration)	Применимая платформа (Applicable Platform)	Дополнительная информация о списке применимых платформ для заданного слабого места.
	Поддержка (Maintenance)	Значимые задачи поддержки, которые необходимо выполнить, такие как уточнение концептов, задействованных в отношениях.
	Отношение (Relationship)	Уточнение деталей отношений между сущностями.
	Пробелы в исследованиях (Research Gap)	Возможности по проведению дальнейших исследований вопросов, связанных со слабым местом, для исследовательского сообщества.
	Терминология (Terminology)	Обсуждение терминологических вопросов, связанных со слабым местом, или уточнения, если нет установленной терминологии, или в случае использования одного и того же ключевого термина в разных случаях.
	Теория (Theoretical)	Описывает слабое место с использованием концептов теории уязвимостей.
	Другое (Other)	
Стратегии минимизации воздействия (MitigationStrategyEnumeration)	Снижение поверхности атаки (Attack Surface Reduction)	
	Усиление защиты компиляция или сборки (Compilation or Build Hardening)	
	Приведение в жизнь путем преобразования (Enforcement by Conversion)	
	Усиление защиты среды (Environment Hardening)	
	Межсетевой экран (Firewall)	

Название перечисления	Возможные значения	Описание значений
Стратегии минимизации воздействия (MitigationStrategyEnumeration)	Проверка ввода (Input Validation)	
	Выбор языка (Language Selection)	
	Библиотеки или фреймворки (Libraries or Frameworks)	
	Ограничение ресурсов (Resource Limitation)	
	Кодировка вывода (Output Encoding)	
	Параметризация (Parameterization)	
	Рефакторинг (Refactoring)	
	Песочница или Jail (Sandbox or Jail)	
	Разделение привилегий (Separation of Privilege)	
Эффективность (EffectivenessEnumeration)	Высокая (High)	Минимизация воздействия зачастую успешна для полного устранения слабого места.
	Средняя (Moderate)	Минимизация воздействия будет препятствовать многим формам слабого места, но не предоставляет полного покрытия слабого места.
	Ограниченная (Limited)	Минимизация воздействия может быть полезна при ограниченных обстоятельствах или применима только к подмножеству потенциальных ошибок данного типа слабого места.
	Случайная (Incidental)	Минимизация воздействия обычно неэффективна и предоставляет защиту случайно, а не надежно.
Эффективность (EffectivenessEnumeration)	Защита в глубину (Defense in Depth)	Минимизация воздействия необязательно устранит слабое место, но поможет минимизировать потенциальный ущерб в результате его использования.
	Нет (None)	
Природа (RelatedNatureEnumeration)	Потомок (ChildOf)	Обозначает наличие связанного слабого места более высокого уровня абстракции.
	Предок (ParentOf)	Обозначает наличие связанного слабого места более низкого уровня абстракции.
	Начинается с (StartsWith)	Используется для обозначения слабого места, входящего в цепочку.
	Может следовать за (CanFollow)	Используется для обозначения слабого места, входящего в цепочку.
	Может предшествовать (CanPrecede)	Используется для обозначения слабого места, входящего в цепочку.

Название перечисления	Возможные значения	Описание значений
Природа (RelatedNatureEnumeration)	Требуется (RequiredBy)	Используется для обозначения слабого места, являющегося частью составной структуры слабого места.
	Требует (Requires)	Используется для обозначения слабого места, являющегося частью составной структуры слабого места.
	Может также быть (CanAlsoBe)	Слабое место, которое в правильной среде или контексте может восприниматься также как рассматриваемое слабое место (необязательно взаимно).
	Пара (PeerOf)	Используется для демонстрации иной схожести с рассматриваемым слабым местом.
Порядки (OrdinalEnumeration)	Первичное (Primary)	Определяет, является ли отношение первичным отношением для заданного слабого места в рамках заданного вида.
Соответствия отображения (TaxonomyMapping-FitEnumeration)	Точное (Exact)	
	CWE абстрактнее (CWE More Abstract)	
	CWE конкретнее (CWE More Specific)	
	Неточное (Imprecise)	
	Перспективное (Perspective)	
Имена таксономий (TaxonomyNameEnumeration)	7 Pernicious Kingdoms	
	19 Deadly Sins	
	Aslam	
	Bishop	
	CLASP	
	Landwehr	
	OMG ASCSM	
	OMG ASCRM	
	OMG ASCMM	
	OMG ASCPEM	
	OWASP Top Ten 2004	
	OWASP Top Ten 2007	
	OWASP Top Ten	
	PLOVER	
	Protection Analysis	
	RISOS	
	Weber, Karger, Paradkar	
	WASC	
	CERT C Secure Coding	
	CERT C++ Secure Coding	
	The CERT Oracle Secure Coding Standard for Java (2011)	

Название перечисления	Возможные значения	Описание значений
Имена таксономий (TaxonomyNameEnumeration)	SEI CERT C Coding Standard	
	SEI CERT C++ Coding Standard	
	SEI CERT Oracle Coding Standard for Java	
	SEI CERT Perl Coding Standard	
	Software Fault Patterns	
Упорядочивание (OrdinalityEnumeration)	Непрямое (Indirect)	Слабое место напрямую не ведет к слабым местам, связанным с безопасностью, но является проблемой качества, которая может упростить введение слабых мест, связанных с безопасностью, или усложнить их обнаружение.
	Первичное (Primary)	Слабое место существует независимо от других слабых мест.
	Проистекающее (Resultant)	Слабое место существует только при наличии других слабых мест.
Заинтересованные лица (StakeholderEnumeration)	Академические исследователи (Academic Researchers)	
	Прикладные исследователи (Applied Researchers)	
	Команды оценки (Assessment Teams)	
	Производители инструментов оценки (Assessment Tool Vendors)	
	Команда CWE (CWE Team)	
	Педагоги (Educators)	
	Разработчики аппаратного обеспечения (Hardware Designers)	
	Информационные провайдеры (Information Providers)	
	Потребители продукта (Product Customers)	
	Производители продукта (Product Vendors)	
	Разработчики ПО (Software Developers)	
	Специалисты по анализу уязвимостей (Vulnerability Analysts)	
	Другие (Other)	

Сведения об авторах

Федорченко (Дойникова) Елена Владимировна, кандидат технических наук, старший научный сотрудник лаборатории проблем компьютерной безопасности, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук (СПб ФИЦ РАН). Область научных интересов: безопасность компьютерных сетей и киберфизических систем, оценивание защищенности компьютерных сетей, принятие решений по реагированию на инциденты информационной безопасности, управление кибербезопасностью, анализ данных. Число научных публикаций в рецензируемых изданиях — более 80.

Котенко Игорь Витальевич, доктор технических наук, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук (СПб ФИЦ РАН). Лауреат премии правительства Санкт-Петербурга за выдающиеся научные результаты в области науки и техники (номинация: электро- и радиотехника, электроника и информационные технологии — премия им. А. С. Попова). Область научных интересов: безопасность компьютерных сетей и киберфизических систем, искусственный интеллект, информационно-телекоммуникационные системы. Число научных публикаций в рецензируемых изданиях — более 700.

Для заметок

ОЦЕНИВАНИЕ ЗАЩИЩЕННОСТИ И ВЫБОР КОНТРМЕР ДЛЯ УПРАВЛЕНИЯ КИБЕРБЕЗОПАСНОСТЬЮ

Подписано в печать 03.12.2021. Формат 70x100 1/16.
Гарнитура Times. Печ. л. 11,5.
Тираж 300 экз. Заказ № 6228.

Издатель – Российская академия наук

Оригинал-макет подготовлен
ООО «Красногорский полиграфический комбинат»

Публикуется в авторской редакции

Отпечатано в ООО «Красногорский полиграфический комбинат»
115093 г. Москва, Партийный переулок д. 1 корп. 58, стр. 1, эт. 1, пом. 1

Издается в соответствии с постановлением Научно-издательского совета
Российской академии наук (НИСО РАН) от 12 февраля 2021 г. № 01
и распространяется бесплатно