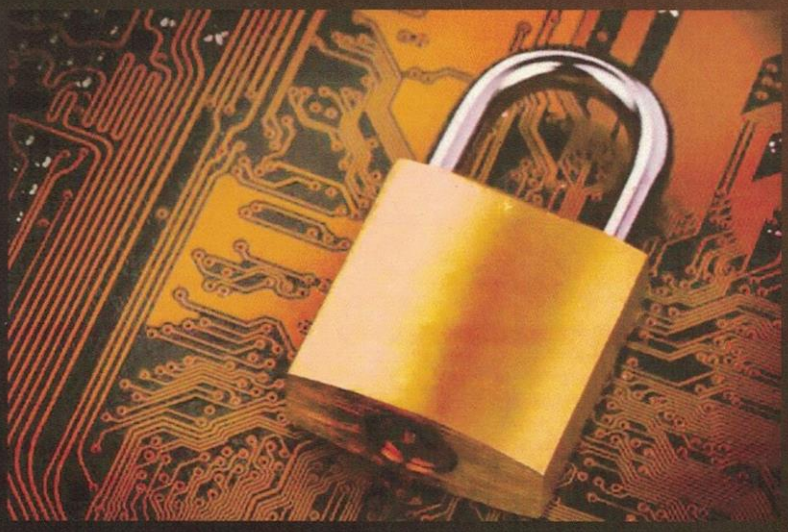
А. Н. Новиков
А. Н. Родионов
А. А. Тимошенко

МОДЕЛИ И МЕТОДЫ

КИБЕРНЕТИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ЛОГИКО-ВЕРОЯТНОСТНОГО ПОДХОДА



Министерство образования и науки Украины
Национальный технический университет Украины
«Киевский политехнический институт»

А. Н. Новиков, А. Н. Родионов, А. А. Тимошенко

Модели и методы кибернетической защиты информационно-коммуникационных систем на основе логико-вероятностного подхода

Монография

Рекомендовано Ученым советом НТУУ «КПИ»

Киев
НТУУ «КПИ»
2015

УДК 004.056
ББК 32.97
Н73

Рекомендовано Ученым советом НТУУ «КПИ»
(Протокол № 9 от 02.11.2015 г.)

Рецензенты:
А.Г. Оксиюк, д-р техн. наук, проф.,
Киевский национальный университет имени Тараса Шевченка

А.К. Юдин, д-р техн. наук, проф.,
Национальный авиационный университет

Ответственный редактор
Н.В. Грайворонский, канд. физ.-мат. наук, доц.,
Национальный технический университет Украины
«Киевский политехнический институт»

Новиков А.Н.

Н73 Модели и методы кибернетической защиты информационно-коммуникационных систем на основе логико-вероятностного подхода: монография / А. Н. Новиков, А. Н. Родионов, А. А. Тимошенко. – Киев: НТУУ «КПИ» Изд-во «Политехника», 2015. – 276 с. – 300 экз.
ISBN 978-966-622-738-9

Представлены современные модели, методы и алгоритмы кибернетической защиты на основе логико-вероятностного подхода, которые используются для решения задач анализа защищенности и синтеза защищенных информационно-коммуникационных систем. Предложены подходы для решения задач параметрического и структурного синтеза систем кибернетической защиты с использованием методов нелинейного программирования. Рассмотрены варианты моделирования возможных последствий изменения параметров и топологии информационно-коммуникационных систем последствий изменения параметров и топологии информационно-коммуникационных систем и их влияния на состояние защищенности системы в целом. Приведены примеры использования разработанных моделей, методов и алгоритмов в практике кибернетической защиты.

Для специалистов, которые занимаются проблемами безопасности, - специалистов по кибернетике, информационным технологиям, для прикладных математиков, ученых, преподавателей, аспирантов и студентов старших курсов университетов по специальностям «Кибербезопасность», «Прикладная математика» и другим, а также для всех, кто желает расширить знания в сфере кибернетической безопасности.

Подано сучасні моделі, методи та алгоритми кібернетичного захисту на основі логіко-ймовірнісного підходу, які застосовують для розв'язання задач аналізу захищеності й синтезу захищених інформаційно-комунікаційних систем. Запропоновано підходи для розв'язання задач параметричного та структурного синтезу систем кібернетичного захисту з використанням методів нелінійного програмування. Розглянуто варіанти моделювання можливих наслідків змін параметрів і топології інформаційно-комунікаційних систем та їх впливу на стан захищеності систем у цілому. Наведено приклади застосування розроблених моделей, методів і алгоритмів у практиці кібернетичного захисту.

Для фахівців, які займаються проблемами безпеки, - фахівців з кібернетики, інформаційних технологій, для прикладних математиків, науковців, викладачів, аспірантів і студентів старших курсів університетів за спеціальностями «Кібербезпека», «Прикладна математика» та іншими, а також для всіх, хто бажає розширити знання у сфері кібернетичної безпеки.

УДК 004.056
ББК 32.97

ISBN 978-966-622-738-9

ВВЕДЕНИЕ

Современный мир характеризуется распространением информационно-коммуникационных технологий, ростом возможностей неограниченного информационного обмена в режиме реального времени. Информационные технологии все глубже проникают в управление государствами, банковское дело, критические технологии, быт обычного человека (интернет вещей). Глобализируется мировое информационное пространство.

В условиях глобализации мира, развития информационных и кибернетических технологий возникает сложная транснациональная проблема роста количества угроз, нарушений и преступности в киберпространстве. Оформились новые направления сомнительной активности, нарушений, манипуляций и преступлений в сети, такие как хактивизм, кибернарушения, киберпреступность, кибертерроризм. Угрозы становятся все более изощренными, появляются новые объекты киберпреступности (критические технологии, персональные данные ведущих руководителей государств, крупных компаний, электронные средства массовой информации, социальные сети и др.).

Украина не является исключением из мировых тенденций: ее государственные структуры, банковская сфера, бизнес и обычные граждане также страдают от махинаций, манипулирований и преступлений в киберпространстве. В настоящее время Парламент Украины рассматривает проект закона «О кибернетической безопасности Украины» как части направления национальной безопасности страны. Цель этого закона – вхождение Украины в мировое законодательное пространство в данном вопросе. Важнейшей составляющей законодательной базы государства при этом является терминологическая оснастка. Фактически первой попыткой ввести терминологию в области кибернетической безопасности в Украине на настоящий момент являются определения основных понятий, введенные в проекте документа «Стратегия обеспечения кибернетической безопасности Украины» [1].

Приведем определения базовых терминов, таких как кибернетическое пространство, кибернетическая угроза, кибернетическая защита и кибернетическая безопасность.

Кибернетическое пространство (киберпространство) – среда, которая возникает в результате функционирования на основе единых принципов и по

общим правилам информационных (автоматизированных), телекоммуникационных и информационно-телекоммуникационных систем.

Кибернетическая угроза (киберугроза) – имеющиеся и потенциально возможные явления и факторы, угрожающие кибернетической безопасности.

Кибернетическая защита (киберзащита) – совокупность мероприятий организационного, нормативно-правового, военного, оперативного и технического характера, направленных на обеспечение кибербезопасности.

Кибернетическая безопасность (кибербезопасность) – состояние защищенности жизненно важных интересов человека и гражданина, общества и государства в киберпространстве.

Ниже будем опираться на эти определения, не вдаваясь в их детальный анализ.

Среди факторов, которые влияют на уровень кибернетической защиты системы (определяют его), можно выделить такие, как качество программного и программно-аппаратного обеспечения, настройка параметров программного и программно-аппаратного обеспечения, архитектура программного обеспечения, пользователи, информационная среда, в которой работает компьютерная система, а также система защиты информации (СЗИ) в организации.

При построении систем кибернетической защиты информации чаще всего применяется подход упреждающей защиты, основанный на предвидении всех возможных, предполагаемых и потенциальных угроз и разработке (проектировании, построении) комплексной системы кибернетической защиты информации (КС КЗИ). Проектирование КС КЗИ может осуществляться с использованием эмпирического подхода методом проб и ошибок, стандартов, нормативных документов, инструкций и руководств, а также формальных математических методов.

При использовании эмпирического подхода одним из недостатков является то, что даже при участии опытного проектировщика прочность спроектированной системы кибернетической защиты бывает или недостаточной, или такой, что превышает необходимый уровень требований, ведет к увеличению ее стоимости.

При использовании подхода к синтезу системы кибернетической защиты с использованием стандартов безопасности основным преимуществом является предоставление разработчику прозрачной технологии синтеза конкретных

систем защиты с гарантированным уровнем безопасности. Недостаток данного подхода – отсутствие единства, даже при четком следовании стандартам и нормативным документам, решение задач синтеза конкретной архитектуры и выбор количественных характеристик систем защиты. Также, как и при эмпирическом подходе, качество и правильность выполнения задач и полученных результатов будут определяться субъективным фактором и зависеть от профессионализма разработчиков. Кроме этого, данные задачи могут иметь неоднозначность решения, допускать несколько различных интерпретаций, оставлять на усмотрение разработчиков вопросы эффективного использования механизмов защиты, учета показателей стоимости, риска, надежности и других факторов.

Применение формальных математических методов синтеза систем кибернетической защиты позволяет уменьшить указанные недостатки. Данный подход заключается в определении политики безопасности, критерия безопасности и модели безопасности ИКС в формальном виде. Далее вводятся дополнительные количественные требования к качеству построения и функционирования СЗИ в виде критериев качества, систем ограничений, включающих математическую модель. Основываясь на этом, решение задачи синтеза осуществляется путем поиска экстремума критерия качества с использованием методов математического программирования (методов линейного булевого программирования (ЛБП), методов сепарабельного программирования, градиентных методов и др.), теории игр, теории принятия решений, методов системного анализа и других.

Значительный вклад в развитие формальных методов синтеза систем защиты информации сделали В. А. Герасименко, П. Д. Зегжда, П. Н. Девянин, А. А. Грушо, Л. Д. Хоффман, Б. Шнаер, В. В. Домарев и др.

Вместе с тем уровень развития формальных подходов синтеза систем защиты в настоящее время является недостаточным и требует дальнейшего развития. В частности, актуальной является разработка моделей, методов и алгоритмов, которые могут быть применены для решения следующих задач:

– анализ защищенности информационно-коммуникационных систем (ИКС) на этапе их проектирования и во время эксплуатации;

– выполнение структурного и параметрического синтеза (выбор характеристик, параметров, архитектуры, топологии, механизмов защиты и пр.), наилучшего с точки зрения уровня защищенности;

– моделирование возможных последствий изменений в системе и их влияние на состояние защищенности системы.

Также важным является построение подходов, которые предоставят возможность применения разработанных моделей, методов и алгоритмов в практике кибернетической защиты.

Книга состоит из девяти глав.

В первой главе рассматривается проблема построения СЗИ в ИКС с открытой архитектурой. Анализируются состав и структура ИКС, существующие подходы к построению защищенных ИКС, а также проблемы, которые при этом возникают. Рассматриваются прикладные модели, использующиеся для построения систем защиты и анализа защищенности ИКС.

Во второй главе на основе исследования условий, особенностей и закономерностей функционирования ИКС с открытой архитектурой с точки зрения обеспечения защиты информации исследуются особенности задачи построения защищенной ИКС. С учетом условий и закономерностей функционирования информационной системы определены моделируемая характеристика, структура и набор параметров разрабатываемой математической модели защищенной ИКС. Обоснованы выбор множества объектов моделируемой системы, а также способ представления в разрабатываемой модели множества угроз информации, реализуемых по отношению к данным объектам.

В третьей главе рассматриваются задачи построения иерархической логико-вероятностной модели защищенной ИКС с открытой архитектурой. С использованием логико-вероятностного метода разработана математическая модель защищенной ИКС с открытой архитектурой. Для этого были построены модели ИКС с двухуровневым и трехуровневым стеками протоколов при одной угрозе информации, проанализированы их особенности и обобщены на случай системы с произвольным стеком протоколов для произвольного множества угроз. Выполнено вычислительное исследование разработанной математической модели, определено, насколько правильно в ней отражены выявленные условия, и закономерностей функционирования информационных систем с открытой архитектурой.

В четвертой главе решается задача оптимального размещения механизмов защиты по уровням стека протоколов взаимодействия компонентов ИКС. Сформулирована задача построения оптимальной СЗИ с учетом ограничений

на стоимость средств защиты как задача нелинейного целочисленного программирования с одним ограничением. Обоснован выбор метода ее решения. Разработан алгоритм решения сформулированной задачи, в котором учитываются выявленные закономерности функционирования механизмов защиты в информационной системе с открытой архитектурой. Выполнено вычислительное исследование разработанного алгоритма, оценены его сходимость и вычислительная эффективность.

В пятой главе приводится модель ИКС с учетом топологии сети, строятся комплексная модель угроз на ИКС, основанная на графе атак, и модель системы защиты. Для оценки уровня защищенности ИКС применяется логико-вероятностный метод анализа безопасности структурно сложных систем (ССС) и строится функция вероятности успешной атаки. Далее описывается метод оценки уровня защищенности ИКС.

Шестая глава посвящена вопросам построения безопасной топологии сети ИКС на основе структурной логико-вероятностной модели, показано влияние топологии сети на уровень защищенности ИКС, предложен подход к построению безопасной топологии сети. Исходя из того, что функция вероятности успешности атаки, рассмотренная в предыдущем разделе, зависит от топологии сети, используется ее значение в качестве критерия для выбора безопасной топологии. Будем считать, что некая топология сети будет более безопасной в сравнении с другими вариантами топологий, если функция вероятности успешности атаки принимает на ней минимальное значение.

Седьмая глава посвящена решению задачи оптимального размещения механизмов защиты в ИКС при ограничении на стоимость СЗИ, где в качестве целевой функции используется модифицированная функция вероятности успешности атаки. Для решения задачи оптимального размещения механизмов защиты применяется алгоритм последовательного анализа вариантов.

В восьмой главе рассматриваются задачи определения экстремальных экономических показателей систем кибернетической защиты. Эти показатели определяются стоимостью механизмов защиты, которые были выбраны на этапе проектирования соответствующей КС КЗИ, затратами на преодоление последствий успешных кибератак злоумышленников, накладными и другими затратами для обеспечения функционирования КС КЗИ при учете ограничений на значение уровня киберзащищенности.

В девятой главе описан логико-вероятностный подход к построению систем кибернетической защиты. Рассмотрены этапы построения защищенной ИКС и логико-вероятностный подход к построению защищенных ИКС.

Выражаем благодарность Издательству «Политехника» НТУУ «КПИ» за плодотворное сотрудничество при подготовке рукописи к изданию. Также выражаем благодарность рецензентам – члену-корреспонденту Академии связи Украины, директору Института компьютерных информационных технологий НАУ, профессору, доктору технических наук А. К. Юдину и заведующему кафедрой киберзащиты и защиты информации факультета информационных технологий КНУ имени Тараса Шевченко, доктору технических наук А. Г. Оксиюку – за содержательные замечания и предложения по повышению качества этой книги.

Авторы выражают искреннюю благодарность за поддержку нашим коллегам по Физико-техническому институту НТУУ «КПИ» – аспирантам Ю. Ю. Боне, А. М. Хныгичевой и О. В. Цыганковой, с которыми обсуждались отдельные разделы этой книги и были учтены их ценные предложения.

1. ПРОБЛЕМА ПОСТРОЕНИЯ СИСТЕМЫ КИБЕРЗАЩИТЫ В ИНФОРМАЦИОННО-КОММУНИКАЦИОННОЙ СИСТЕМЕ С ОТКРЫТОЙ АРХИТЕКТУРОЙ

В данной главе рассматривается проблема обеспечения кибернетической безопасности в ИКС с открытой архитектурой. Анализируются состав и структура ИКС, существующие подходы к построению защищенных ИКС, а также проблемы, которые при этом возникают. Рассматриваются математические модели, использующиеся для построения систем кибернетической защиты и анализа защищенности ИКС.

1.1. Информационно-коммуникационные системы как объект киберзащиты

Под ИКС будем понимать следующее: ИКС, или информационно-коммуникационная технология – это совокупность информационных систем, сетей и каналов передачи информации, способов коммуникации и управления информационными потоками. Как можно заметить, ИКС является составным элементом киберпространства.

Решение о создании ИКС в организации или на предприятии принимается, исходя из набора целей, важных для последующей деятельности организации. Создание подобной системы ради нее самой может происходить во время выполнения определенных исследований или проведения модельных экспериментов (имитационного моделирования).

Информационно-коммуникационная система может создаваться для решения следующих задач [2]:

– информационная система для организационного управления (предназначена для автоматизации функций управляющего персонала как государственных органов, промышленных предприятий, так и других объектов (гостиниц, банков, магазинов и пр.));

– информационная система для управления технологическим процессом (предназначена для автоматизации функций производственного персонала по контролю и управлению производственными операциями);

– информационная система автоматизированного проектирования (предназначена для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов при создании новой техники и технологий);

– интегрированная (корпоративная) информационная система (используется для автоматизации всех функций организации и охватывает весь цикл работы от планирования до реализации продукции);

– облачные вычисления (cloud computing), предоставляющие «готовые» информационные системы вида: Программное обеспечение как услуга (Software-as-a-Service (SaaS)), Платформа как услуга (Platform as a Service (PaaS)) и Инфраструктура как услуга (Infrastructure-as-a-Service (IaaS)).

Исходя из определения ИКС, ее состав с помощью структурной декомпозиции можно изобразить в виде следующей диаграммы (рис. 1.1), где каждый из объектов диаграммы допускает детальное разделение на ряд составляющих.

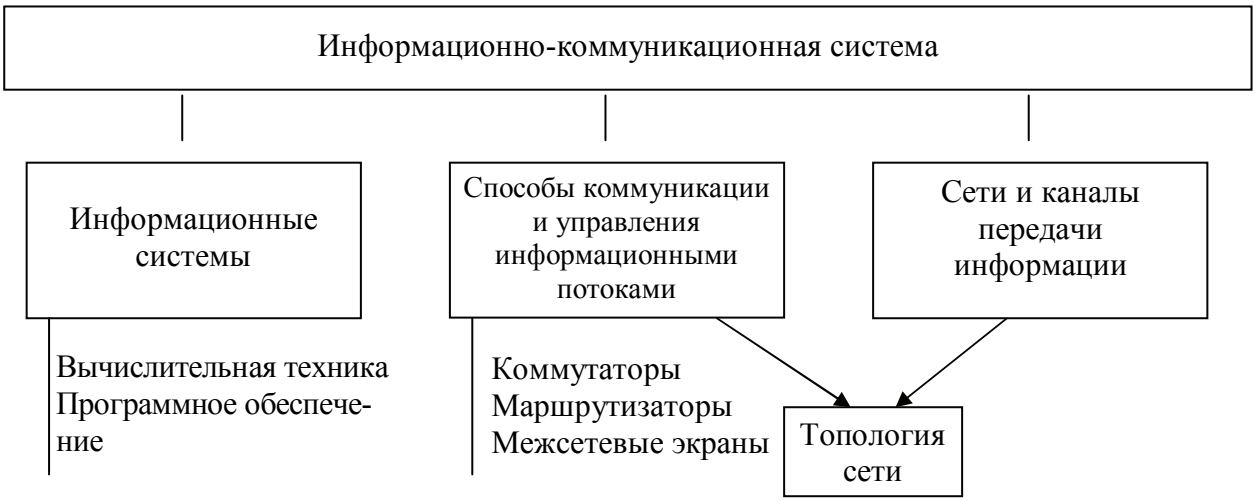


Рис. 1.1. Состав ИКС

Информационные системы ИКС. Для начала остановимся на информационной составляющей ИКС.

Приведем определение информационной системы. Информационная система – это организационно-техническая система, которая реализует технологию обработки информации с помощью средств вычислительной техники и программного обеспечения [3].

Классически в информационной системе выделяют четыре уровня [4]:

1. Уровень сети – отвечает за взаимодействие узлов ИКС.

2. Уровень операционных систем – отвечает за обслуживание программного обеспечения, которое реализует более высокие уровни, и его взаимодействие с оборудованием.

3. Уровень систем управления базами данных (СУБД) – отвечает за хранение и обработку данных.

4. Уровень прикладного программного обеспечения – включает прикладные компоненты и интерфейс взаимодействия с пользователем.

Вместе с тем при проектировании программного обеспечения выделяют следующие варианты архитектуры его построения и функционирования [5]:

– одноуровневая – клиентская, доступ к другим ресурсам и сервисам сети не требуется;

– двухуровневая – клиент-серверная, преимущественно используется в программах, которые напрямую обращаются к СУБД, обычно требуется доступ к сервисам сети;

– трехуровневая и многоуровневая – распределенная, требуется доступ к нескольким сервисам локальной сети, может потребоваться доступ к сервисам, которые находятся за пределами организации;

– сервисно ориентированная архитектура (облачные сервисы и программное обеспечение) – вариант многоуровневой архитектуры, которая строится с использованием технологий Web 2.0.

Почти каждая ИКС является уникальной и создается, соответственно, с учетом потребностей организации. Она будет содержать набор программного обеспечения, которое будет работать на всех четырех уровнях, и может строиться с использованием всех вариантов архитектуры.

Принятая в организации структура бизнес-процессов и бизнес-правил, а также архитектура многоуровневых приложений во многом будут обуславливать архитектуру информационной системы, ее состав, характеристики и влиять на топологию сети.

Особый интерес представляют многоуровневые приложения, которые интегрируют несколько разных сервисов и могут создавать несколько информационных потоков, необходимых для их функционирования.

Среди основных информационных систем и сервисов, которые являются общими для ИКС, можно выделить следующие (табл. 1.1) [6, 7].

Таблица 1.1

Основные сервисы информационно-коммуникационных систем

Сервисы	Технологии
Сетевые устройства	Маршрутизаторы, коммутаторы, шлюзы, балансирование нагрузки, точки доступа, модемы, фареволы…
Сетевые сервисы	DNS, DHCP, WINS, NFS, Printing …
Сервисы межсетевой фильтрации трафика и доступа	Межсетевые экраны периметра, демилитаризованной зоны, внутренние сегменты сети, прокси-серверы, системы обнаружения и предотвращения вторжений (IDS/IPS) …
Сервисы для централизованного управления и мониторинга корпоративной сетью	Microsoft Active Directory, Oracle Identity Manager, IBM Tivoli Identity Manager, LDAP, SNMP, Cisco Works, Microsoft System Center, Tivoli …
Системы управления базами данных	Microsoft SQL Server, Oracle, DB2, MySQL …
Устройства хранения данных	Direct-attached storage (DAS), network attached storage (NAS), storage area network (SAN)
Серверы приложений	Java EE, .NET Framework, COM+ and Message Queuing Services (MSMQ) …
Веб-сервисы и веб-серверы	Apache, GlassFish, IIS, WebSphere, WebLogic …
Сервер электронной почты	SMTP, POP3, IMAP …
Операционные системы на хостах сети	Windows, Linux, Solaris, AIX …
Сервисы для совместной работы	SharePoint, Lotus Notes, Lotus Connection, Google Docs, Collaboration server …
Сервисы обеспечения удаленного доступа	Site-to-site virtual private network (VPN), Client VPN, RRAS, Internet authentication service (IAS)
Антивирусы	Symantec, Kaspersky, ESET NOD32, Dr.Web …
Безопасность	Обеспечение целостности, доступности и конфиденциальности, разделения доступа, идентификация и аутентификация, Network Access Control (NAC), Single Sign On (SSO), LDAP, PKI

Перечень сервисов в табл. 1.1 не является полным. Каждый из приведенных сервисов или служб предоставляет пользователям или другим информационным системам доступ к множеству своих функций. Соответственно, для каждого из них необходимо определить перечень пользователей и посторонних сервисов, прав их доступа, планировать оптимальное размещение в сети с учетом взаимозависимости и информационных потоков.

Ошибки при выборе архитектуры информационной системы, структуры ее развертывания, в настройках и другие могут привести к неработоспособности всей ИКС.

Средства коммуникации и управления информационными потоками ИКС. Перейдем к рассмотрению коммуникационной составляющей и способов управления информационными потоками в ИКС.

Важной характеристикой ИКС является топология сети. Топология – это конфигурация графа, вершинам которого соответствуют узлы (объекты) сети (серверы, компьютеры и, возможно, устройства коммутации), а ребрам – связи между ними. Средства коммутации и управления информационными потоками вместе с сетями и каналами передачи информации будут образовывать топологию сети.

Различают физическую и логическую топологию.

Физическая топология описывает конфигурацию физических (электрических, оптических и т. д.) связей между компьютерами (сети и каналы передачи информации на схеме), а логическая топология – конфигурацию возможных маршрутов обмена сообщениями (средства коммуникации и управления информационными потоками на схеме). Физическая и логическая топологии некоторой сети могут существенно отличаться [8].

В нормативных документах о технической защите информации Украины (НД ТЗИ), используется термин более широкий, чем ИКС, а именно автоматизированная система (АС).

В соответствии с НД ТЗИ 2.5-005-99 [9] АС представляет собой организационно-техническую систему, которая объединяет операционную систему (ОС), физическую среду, персонал и обрабатываемую информацию. Каждая из этих составляющих непосредственно будет влиять на общий уровень защищенности, иметь набор характеристик, требований относительно настройки и организационных требований.

Также в этом документе по совокупности характеристик АС (конфигурация аппаратных средств ОС и их физическое размещение, количество разнообразных категорий обрабатываемой информации, количество пользователей и категорий пользователей) выделено три иерархических класса АС, требования к функциональному составу комплекса средств защиты (КСЗ) которых существенно отличаются.

Класс «1» – одномашинный однопользовательский комплекс, который обрабатывает информацию одной или нескольких категорий конфиденциальности.

Существенные особенности: в каждый момент времени с комплексом может работать только один пользователь, хотя в общем случае лиц, которые имеют доступ к комплексу, может быть несколько, но все они должны иметь одинаковые полномочия (права) относительно доступа к информации, которая обрабатывается.

Пример – автономная персональная ЭВМ, доступ к которой контролируется с использованием организационных мероприятий.

Класс «2» – локализованный многомашинный многопользовательский комплекс, который обрабатывает информацию разных категорий конфиденциальности.

Существенное отличие от предыдущего класса – наличие пользователей с разными полномочиями по доступу и/или технических средств, которые могут одновременно осуществлять обработку информации разных категорий конфиденциальности.

Пример – локальная вычислительная сеть.

Класс «3» – распределенный многомашинный многопользовательский комплекс, который обрабатывает информацию разных категорий конфиденциальности.

Существенное отличие от предыдущего класса – необходимость передачи информации через незащищенную среду или, в общем случае, наличие узлов, которые реализуют разную политику безопасности.

Пример – глобальная сеть.

Таким образом, ИКС может содержать, начиная от нескольких хостов, находящихся в пределах одной комнаты, и заканчивая большими распределенными корпоративными сетями, подключенными к Интернет. Понятно, что чем больше ИКС, тем больше параметров необходимо учитывать для обеспечения ее безопасности.

Как результат, ИКС будет являться частью киберпространства и представлять собой структурно сложную территориально распределенную гетерогенную среду, в которой необходимо обеспечивать интероперабельность, масштабируемость и кибербезопасность.

1.2. Анализ особенностей ИКС с открытой архитектурой

Одним из современных направлений развития информационных технологий, определяющим эффективность ИКС всех уровней и назначений, в совокупности составляющих единое кибернетическое пространство, выступает технология систем с открытой архитектурой (открытых систем). Основным достоинством ее использования является возможность сохранения сделанных ранее инвестиций. Сущность технологии открытых систем состоит в обеспечении возможности переносимости прикладных программ между различными платформами и обеспечении, таким образом, взаимодействия систем, входящих в единое киберпространство, друг с другом. Эта возможность достигается за счет использования международных стандартов на все программные и аппаратные интерфейсы между компонентами систем. Согласно определению Национального института стандартов (NIST) США [10]: «Открытая система – это система, которая способна взаимодействовать с другой системой посредством реализации международных стандартных протоколов. Открытыми системами являются как конечные, так и промежуточные системы. Однако открытая система не обязательно может быть доступна другим открытым системам. Эта изоляция может быть обеспечена или путем физического отделения, или путем использования технических возможностей, основанных на защите информации в компьютерах и средствах коммуникаций».

Защита от кибернетических угроз в той или иной форме необходима всем пользователям ИКС, функционирующих в кибернетическом пространстве, в том числе и пользователям открытых систем. При этом уровень необходимой защиты для разных пользователей и разных систем меняется в широком диапазоне. Вообще говоря, возникает противоречие между степенью защиты и легкостью доступа к системе. С одной стороны, наличие средств защиты потенциально может влиять на производительность системы в целом, на удобство ее использования, на взаимодействие приложений и общее управление системой. С другой стороны, такие свойства открытых систем, как интероперабельность (возможность обеспечить совместную работу с другими прикладными системами на локальных и удаленных платформах) и мобильность приложений (возможность переноса прикладных систем, разработанных

должным образом, с минимальными изменениями на широкий диапазон систем), несут в себе источник дополнительных уязвимостей, например, опасность внесения вирусов или «троянских коней», или облегчение неавторизованного доступа.

Из сказанного ясно, что обеспечение кибернетической безопасности ИКС, реализованных в соответствии с технологией открытых систем (ИКС с открытой архитектурой), представляет собой особую проблему, весьма далекую от окончательного решения. В настоящее время частично разработаны и продолжают разрабатываться стандарты, направленные на обеспечение защиты информации в открытых системах, и соответствующие механизмы защиты [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. В Украине приняты и действуют нормативные документы [22, 23], регламентирующие порядок организации работ в области защиты информации, обрабатываемой в ИКС, от НСД и устанавливающие критерии оценки защищенности обрабатываемой информации. К сожалению, в данных документах изложены только требования к функциональным услугам безопасности (функциям, которые должны быть реализованы в СЗИ, обрабатываемой в ИКС, для обеспечения их кибернетической защиты), никаких требований ни к механизмам реализации услуг, ни к специфике реализации различных механизмов защиты в ИКС с открытой архитектурой не предъявляется. Именно решению проблемы построения (синтеза) СЗИ для обеспечения кибернетической безопасности ИКС с открытой архитектурой посвящена данная книга.

Задача обеспечения взаимодействия объектов (компьютеров), входящих в состав ИКС и объединенных в сеть, включает в себя решение множества проблем – выбор способа адресации объектов в сети и согласование электрических сигналов при установлении электрической связи, обеспечение надежной передачи данных и обработка сообщений об ошибках, формирование отправляемых и интерпретация полученных сообщений и т. д. Данная задача обычно решается путем декомпозиции на несколько частных подзадач, для решения каждой предназначается некоторый модуль. Все множество модулей, решающих подзадачи, разбивается на иерархически упорядоченные группы – уровни. Для каждого уровня определяется набор функций/запросов, с которыми к модулям данного уровня могут обращаться модули вышележащего уровня для решения своих задач. Такой формально определенный набор функций, выполняемых данным уровнем для вышележащего уровня,

а также форматы сообщений, которыми обмениваются два соседних уровня в ходе своего взаимодействия, называется интерфейсом.

Правила взаимодействия двух объектов в сети могут быть описаны в виде набора процедур для каждого из уровней. Такие формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах сети, называются протоколами. Согласованный набор протоколов разных уровней, достаточный для организации межсетевого взаимодействия, называется стеком протоколов.

С точки зрения математической логики [84] понятие стека протоколов можно представить следующим образом. Имеется некий язык D, позволяющий описать взаимодействие между двумя любыми объектами сети. Данный язык может быть декомпозирован (разложен) на семейство языков D_1, D_2, ..., D_n таким образом, что язык D_i зависит только от словоформ языка D_{i-1}. В этом случае взаимодействие между объектами сети может быть описано наборами слов B_1, ..., B_n в языках D_1, D_2, ..., D_n таким образом, что описание B_i синтаксически зависит только от набора B_{i-1}. В этом случае можно говорить об иерархической декомпозиции описания взаимодействия между двумя объектами на n уровней B_1, ..., B_n, где уровень B_i непосредственно зависит только от уровня B_{i-1} (хотя опосредованно от всех остальных низлежащих уровней). Совокупность уровней B_1, ..., B_n представляет собой упорядоченный набор (стек) протоколов взаимодействия, а входящие в нее элементы B_i, $i = 1, \ldots, n$ – протоколы взаимодействия соответствующих уровней.

Международная организация по стандартам (International Standards Organization, ISO) разработала модель, которая четко определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень. Эта модель называется моделью взаимодействия открытых систем (ВОС) ISO [25]. В данной модели взаимодействие делится на семь уровней. Каждый уровень отвечает за обеспечение одного определенного аспекта взаимодействия и поддерживает интерфейсы с выше- и нижележащими уровнями.

На основании запроса приложения к прикладному уровню программное обеспечение данного уровня формирует сообщение (обычно называемое пакетом или кадром) стандартного формата, в которое помещает служебную

информацию (заголовок) и, возможно, передаваемые данные. Затем это сообщение направляется уровню представления. Уровень представления добавляет к сообщению свой заголовок и передает результат вниз сеансовому уровню, который, в свою очередь, добавляет свой заголовок и т. д. Наконец, сообщение достигает самого низкого, физического уровня, который действительно передает его по сетевому каналу связи.

Когда сообщение по сети поступает на другой объект (компьютер), оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует, обрабатывает и удаляет заголовок своего уровня, выполняет соответствующие данному уровню функции и передает сообщение вышележащему уровню.

В [8, 25, 26, 27, 28, 29] приведено описание функций, реализуемых различными уровнями стека протоколов эталонной модели ВОС, а в [31] – характеристика наиболее широко используемых стеков сетевых протоколов и их соотношение с уровнями стека протоколов эталонной модели ВОС.

Анализ приведенных в [8, 25, 26, 27, 28, 29] описаний функций, реализуемых различными уровнями стека протоколов эталонной модели ВОС, описаний наиболее широко используемых стеков сетевых протоколов и их соответствия эталонной модели ВОС позволяет выделить следующие особенности, касающиеся обеспечения кибернетической безопасности:

– в связи с полной открытостью описаний межуровневых протоколов взаимодействия B_i имеется возможность реализации злоумышленником средств, реализующих протокол любого уровня;

– поскольку на i-м уровне стека протоколов B_i обрабатывается вся информация, которая порождена на более высоких уровнях B_{i+1}, ..., B_n, а сам протокол B_i зависит от протоколов всех низлежащих уровней B_1, ..., B_{i-1}, кибернетические угрозы, реализованные на более низких уровнях стека протоколов, имеют более серьезные последствия.

1.3. Теория защиты информации

Теория защиты информации – это наука об общих принципах и методах построения защищенных ИКС.

В настоящее время в теории защиты информации для анализа и синтеза систем кибернетической защиты применяют два подхода – формальный и

неформальный (описательный). Формальный подход состоит в определении политики безопасности, критериев безопасности и моделей безопасности ИКС в формальном виде. Этот подход находится на этапе становления и не может удовлетворить все требования, возникающие во время исследования и создания систем кибернетической защиты. Поэтому его дополняют традиционным неформальным (описательным) подходом.

Неформальный подход теории защиты информации представляет собой описание методов и механизмов, которые используются для защиты информации.

Следует отметить, что теория защиты информации как наука находится еще на стадии развития. Рассмотрим несколько базовых проблем теории защиты информации, которые в настоящее время являются актуальными.

Важной проблемой теории защиты информации является проблема сложности задачи изучения (анализа) СЗИ. В современной теории защиты информации эту проблему решают, применяя метод иерархической декомпозиции сложных систем. С помощью этого метода общую сложную систему распределяют на иерархические уровни.

Верхний уровень иерархии занимает политика безопасности, второй уровень – системы поддержки политики безопасности (мандатный контроль, аудит и т. д.), третий – механизмы защиты (криптографические протоколы и алгоритмы, межсетевые экраны и пр.), четвертый – реализация механизмов безопасности.

Изучение этих подсистем проводят с использованием специфических для каждого уровня иерархии методов анализа.

Следующей проблемой теории защиты информации является проблема построения (синтеза) «гарантированно защищенной системы». Эта проблема заключается в противоречии между требованиями к гарантированности и принципиальной невозможностью построить «гарантированно защищенную систему» в классе открытых систем. В теории открытых систем проблему гарантированной безопасности относят к алгоритмически неразрешимым проблемам.

Алгоритмически неразрешимой проблемой является класс задач, для которого нельзя предложить никакого единственного алгоритма, который решал бы все задачи указанного класса.

Доказательное обоснование отсутствия гарантированной безопасности в открытых системах предоставляет, например, теорема о невозможности решения задачи обеспечения безопасности произвольной системы в общем случае при условии недетерминированного задания прав на доступ, сформулированная в работе М. Харрисона, В. Руззо, Дж. Ульмана [30].

Решение этой проблемы проводится путем декомпозиции общей исходной проблемы гарантированной защиты информации в компьютерных системах на совокупность двух задач. Первая из этих задач состоит в корректной формулировке политики безопасности, а вторая – в построении (синтезе) системы кибернетической защиты информации, которая гарантированно поддерживает политику безопасности [4].

В большинстве случаев (а иногда это условие является обязательным) во время формулирования политики безопасности используются разнообразные стандарты (отраслевые, общие, государственные, международные).

Как уже указывалось во введении, преимуществом подхода к синтезу системы кибернетической защиты информации с использованием стандартов безопасности является предоставление разработчику прозрачной технологии синтеза архитектуры конкретных СЗИ с гарантированным уровнем безопасности. Недостатком является отсутствие единственности решения задач синтеза конкретной архитектуры и выбора количественных характеристик СЗИ.

На усмотрение разработчиков остаются вопросы относительно эффективного использования механизмов защиты, учета показателей стоимости, риска, надежности и других [32].

Для частичного устранения приведенных недостатков задачи синтеза систем защиты могут решаться с использованием математической теории оптимизации, теории системного анализа и принятия решений [4], а именно, для формализации задачи синтеза и поиска ее оптимального решения вводятся дополнительные количественные требования к качеству построения и функционирования системы кибернетической защиты информации в виде критериев качества, систем ограничений, которые включают математическую модель. Тогда решение задачи осуществляется путем поиска экстремума критерия качества с использованием методов математического программирования.

1.4. Анализ особенностей известных классических моделей безопасности

Рассмотрим математические модели безопасности, которые могут быть применены во время разработки ИКС или использованы для синтеза системы кибернетической защиты информации, а также модели, с помощью которых можно делать оценку уровня защищенности ИКС.

Решение задачи обеспечения кибернетической безопасности ИКС невозможно без наличия и использования соответствующих моделей. В литературе, посвященной вопросам защиты информации, обрабатываемой в ИКС, приведено описание целого ряда моделей функционирования СЗИ. Проведем их краткий анализ.

Согласно [33] системная классификация данных моделей может быть проведена на основании следующих критериев:

– способ моделирования, согласно которому все модели могут быть разделены на аналитические – представляемые в виде определенных аналитических или логических зависимостей, позволяющих определять необходимые характеристики путем выполнения необходимых вычислений по данным зависимостям; и статистические – представляемые в виде определенного аналога, отображающего те или иные зависимости реальной системы, причем сами характеристики системы определяются путем соответствующих многократных испытаний (например, с использованием известного метода Монте-Карло);

– характер поведения моделируемой системы (точнее говоря, степень доверия к тем или иным показателям и зависимостям), согласно которому все моделируемые системы могут быть разделены на детерминированные (со строго определенными зависимостями между различными показателями) и стохастические (с нестрого определенными, даже случайными или определяемыми случайными факторами зависимостями между различными показателями);

– масштаб модели, согласно которому модели могут быть общими – позволяющими определять некоторые обобщенные характеристики моделируемой системы и частичными – позволяющими определять только некоторые локальные характеристики.

Визуально системная классификация моделей СЗИ, обрабатываемой в ИКС, может быть представлена в виде, приведенном на рис. 1.2 [33].

Согласно приведенным классификационным признакам проанализируем, модели какого типа потенциально могут быть применены при построении СЗИ для решения задачи обеспечения кибернетической безопасности ИКС с открытой архитектурой.

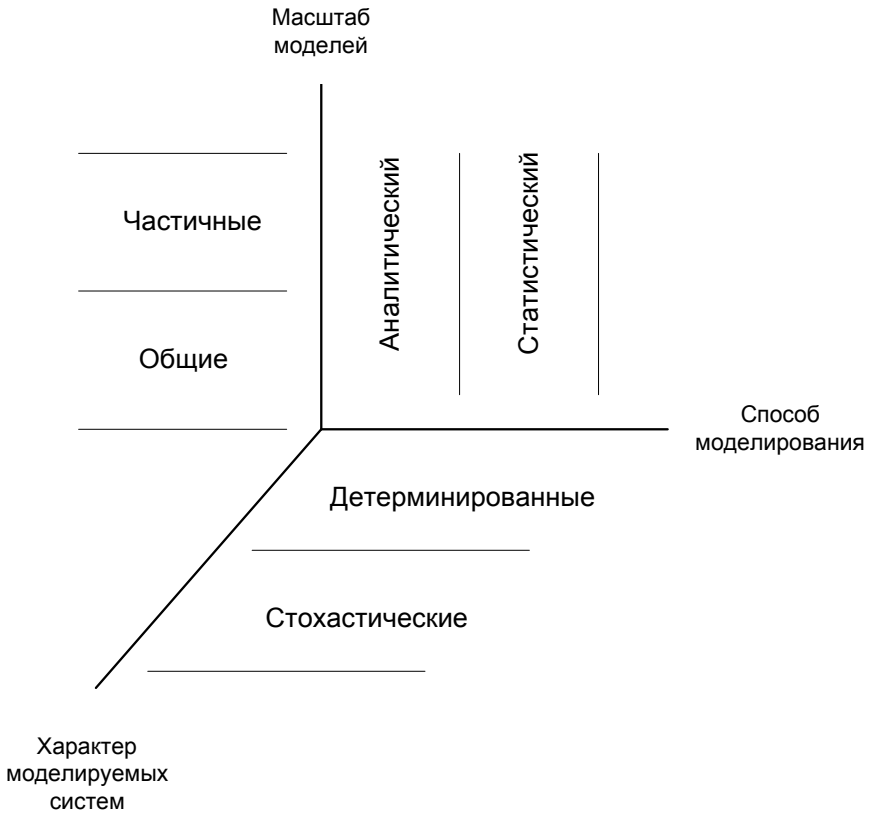


Рис. 1.2. Системная классификация моделей СЗИ

Поскольку задача построения СЗИ предполагает выбор множества механизмов защиты для всей ИКС, частичные модели, позволяющие определять только некоторые локальные характеристики моделируемой системы, не могут быть применены для ее решения, т. е. используемая модель должна относиться к классу общих.

Поскольку задача построения СЗИ предполагает анализ различных вариантов и выбор из определенного множества механизмов защиты в ИКС, используемая модель должна обеспечивать возможность определения тех или иных системных характеристик как функции от механизмов защиты, реализованных в системе. К сожалению, статистические модели не обеспечивают такую возможность в явном виде, поэтому используемая модель должна относиться к классу аналитических.

Выбор модели с точки зрения характера поведения моделируемой системы представляет собой более сложную задачу. Для этого целесообразно проанализировать общие принципы построения аналитических моделей функционирования СЗИ для детерминированных и стохастических систем.

Как уже было сказано выше, главной идеей, положенной в основу построения аналитических моделей функционирования СЗИ для детерминированных систем, является идея описания процесса функционирования (траектории) системы с использованием определенных аналитических или логических зависимостей, при этом может использоваться различный математический аппарат.

1.4.1. Классические модели безопасности

Хотя формальная теория защиты информации начала развиваться сравнительно недавно, сегодня есть много математических моделей, которые описывают разные аспекты безопасности и предоставляют доказательную теоретическую базу для построения СЗИ [4].

Такие классические модели, как Харрисона–Руззо–Ульмана, Take-Grant, Бела–ЛаПадула, Биба и др., описанные и исследованные в работе [34], предоставляют возможность формального доказательства защищенности информации в ИКС.

Модель матрицы доступов Харрисона–Рузо–Ульмана (HRU) [34] построена с использованием аппарата теории отношений. В данной модели система защиты представляется как конечный автомат, а ее функционирование рассматривается с точки зрения изменений в матрице доступа. В развивающую подходы модели HRU модель типизированной матрицы доступа дополнительно введена концепция типов.

Модель распределения прав доступа Take-Grant [34, 35] построена с использованием аппарата теории графов. В данной модели состояния системы и переходы между состояниями описываются при помощи операций преобразования графа доступов.

Классическая модель Белла–ЛаПадулла и модели, основанные на ней [35, 95] (модель безопасного перехода, модель с уполномоченными субъектами, модель совместного доступа, модель безопасного перехода для системы совместного доступа, модель совместного доступа с уполномоченными субъектами),

построены с использованием аппаратов теории отношений и решеток. В данных моделях вводятся понятия функции безопасности и решетки уровней безопасности, которые определяют все разрешенные отношения между сущностями системы. В моделях, развивающих классическую, вводятся также различные дополнительные правила, регламентирующие возможность осуществления переходов, изменяющих уровни безопасности сущностей системы.

Модель Trusted Mach [36, 95] построена с использованием теории предикатов. Она представляет собой модель состояний и событий, в которой используется оператор следующего значения, определяющий новые значения переменных состояния после произошедших событий.

Модель Law-Water-Mark (LWM) [34] построена с использованием теории отношений. Данная модель описывает порядок безопасного функционирования системы в случае, когда по запросу субъекта ему всегда необходимо предоставлять доступ для записи в объект.

Разница между перечисленными (а также другими, относящимися к данному классу) моделями состоит в используемом математическом аппарате, в применяемых для определения возможности предоставления или отказа в доступе к объекту правилах, а также в функциях, эти правила реализующих. Несмотря на формальное различие данных моделей, общими между ними является то, что во всех из них тем или иным образом вводится множество субъектов (пользователей) U, объектов (защищенных ресурсов) R, полномочий доступа L, запросов доступа $Q(U,R)$, правил предоставления доступа или отказа в доступе $F(U,R,L)$.

К сожалению, ни одна из приведенных моделей не оперирует такими понятиями, как угроза или механизм защиты, поэтому они представляют скорее теоретический интерес с точки зрения возможности формального доказательства защищенности информации в ИКС, чем практический – с точки зрения возможности их применения при синтезе СЗИ [84, 95].

1.4.2. Модель системы защиты с полным перекрытием

Согласно модели Хоффмана [37] система безопасности с полным перекрытием должна иметь по крайне мере один способ обеспечения безопасности на каждом возможном пути проникновения в систему. У модели точно определяется каждая область, которая требует защиты, оцениваются средства обеспечения безопасности с точки зрения их эффективности и их вклад

в обеспечение безопасности во всей вычислительной системе. Считается, что несанкционированный доступ (НСД) к каждому из наборов защищаемых объектов O связан с некоторой «величиной ущерба», причем этот ущерб может (или не может) быть определен количественно.

С каждым объектом, требующим защиты, связывается некоторое множество действий, к которым может прибегнуть злоумышленник для получения НСД к объекту. Все потенциальные злоумышленные действия по отношению ко всем объектам рассматриваются как набор угроз T, направленных на нарушение безопасности. Основной характеристикой набора угроз является вероятность проявления каждого из злоумышленных действий.

Множество отношений объект-угроза образуют двудольный граф, в котором ребро $<t_i,o_j>$ существует тогда и только тогда, когда t_i является средством получения доступа к объекту o_j. При этом связь между угрозами и объектами не является связью типа «один к одному», угроза может распространяться на любое число объектов, и объект может быть уязвим со стороны более чем одной угрозы.

Завершает модель третий набор, включающий средства (механизмы) обеспечения безопасности M, которые используются для защиты информации в вычислительной системе. Идеально каждый механизм m_k должен устранять некоторое ребро $<t_i,o_j>$.

Набор M механизмов обеспечения безопасности преобразуют двудольный в трехдольный граф (рис. 1.3). В «защищенной» системе все ребра представляются в форме $<t_i,m_k>$ и $<m_k,o_j>$. Любое ребро в форме $<t_i,o_j>$ определяет незащищенный объект. Один и тот же механизм защиты может перекрывать более одной угрозы и/или защищать более одного объекта.

На основе данной модели построена, например, базовая система безопасности Клементса, представляющая собой пятикортежный набор

$$S = \{O,T,M,V,B\},$$

где: O – набор защищаемых объектов;

T – набор угроз;

M – набор механизмов защиты;

V – набор уязвимых мест – отображение $T \times O$ на набор упорядоченных пар $v_i = <t_i,o_j>$, представляющих собой пути реализации кибернетических угроз (пути проникновения);

B – набор барьеров – отображение $V \times M$ или $T \times O \times M$ на набор упорядоченных троек $b_i = <t_i, o_j, m_k>$, представляющих собой точки, в которых требуется осуществить защиту (реализовать механизмы защиты информации).

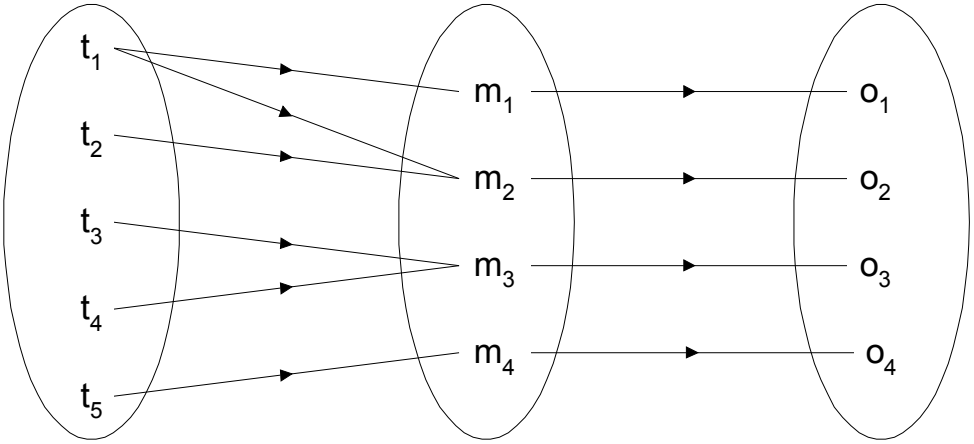


Рис. 1.3. Модель системы безопасности с полным перекрытием

Система с полным перекрытием – это система, в которой имеются средства защиты на каждый возможный путь реализации кибернетических угроз. В такой системе $<t_i, o_j> \in V$ предусматривает $<t_i, o_j, m_k> \in B$. Если это условие не соблюдено, то o_j не защищен для некоторого j.

Как видно, данные модели уже оперируют понятиями кибернетических угроз и механизмов защиты, то есть с их использованием можно выполнять синтез СЗИ. Кроме того, модели данного типа обеспечивают возможность получения численной оценки надежности обеспечения защиты, что позволяет при построении СЗИ минимизировать накладные расходы для реализации заданного уровня безопасности, а при анализе СЗИ – оценить вероятность преодоления системы защиты и оценить степень ущерба системе в случае преодоления системы защиты.

Важно подчеркнуть [95], что модель системы безопасности с полным перекрытием непосредственно не специфицирует модель предотвращения кибернетических угроз механизмами защиты, включенными в СЗИ, а определяет только общий подход к синтезу и анализу.

Таким образом, на основании проведенного анализа можно сделать вывод о том, что для решения задачи построения СЗИ с целью обеспечения кибернетической безопасности ИКС с открытой архитектурой наилучшим является использование моделей, разработанных с учетом принципов построения модели

системы безопасности с полным перекрытием и относящихся к классу общих аналитических моделей для стохастических систем. При этом в разработанных моделях должны учитываться факт функционирования ИКС в условиях риска при неизвестных законах и числовых характеристиках распределения кибернетических угроз, различный ущерб от реализации различных кибернетических угроз, а также особенности функционирования механизмов защиты в таких ИКС. Модели должны обеспечивать возможность оценки вероятности предотвращения заданного множества кибернетических угроз выбранными механизмами защиты, а также соотнесения данной вероятности со стоимостью средств защиты.

1.4.3. Модели вероятностного (стохастического) характера

Данные модели рассматривают кибератаки с разными последствиями (например, нарушение конфиденциальности, нарушение целостности информации, DoS-атаки) и способны передать динамику поведения системы в ответ на атаку. При этом атака злоумышленника и ответ системы моделируются как случайный процесс.

Считается, что система может находиться в конечном наборе существенных с точки зрения безопасности состояний. Моделирование начинается с выделения этих состояний и построения диаграммы переходов между ними. Авторы [38] также предлагают общую диаграмму переходов, которая характеризует поведение систем, стойких к атакам.

Далее строится собственно стохастическая модель поведения злоумышленника и ответов системы. Злоумышленник не владеет полными знаниями о системе, поэтому эффективность атаки заранее точно неизвестна. Аналогично система (разработчик/владелец/пользователь) не имеет точной информации относительно типа, частоты, интенсивности и продолжительности атак. На следующем этапе работы необходимо описать события, которые вызывают переходы между состояниями в терминах вероятностей и функций распределения.

Для того чтобы спланировать свои действия, злоумышленник затрачивает определенное время и усилия. Время/усилия моделируются как случайные величины с определенным распределением, например экспоненциальным, гиперэкспоненциальным, Вейбулловским и т. д., или являются константами. При обнаружении вторжения система может прибегать к определенным действиям в ответ – для этого она тоже потратит определенное время и усилия

(ресурсы), которые в рамках данной модели рассматриваются как случайные величины с определенной функцией распределения.

Рассмотрим случайный процесс $\{X(t), t \geq 0\}$ с дискретным пространством состояний S, которые соответствуют состояниям системы, определенным на первом шаге моделирования. Время пребывания системы в каждом состоянии может иметь разное распределение, отличное от экспоненциального, поэтому этот случайный процесс будет полумарковским. Для анализа полумарковского процесса нужно задать два набора параметров:

– среднее время нахождения h_i в состоянии $i, i \in S$;

– вероятность перехода между различными состояниями $p_{ij}\ i, j \in S$.

Для количественной характеристики атрибутов безопасности (конфиденциальность, целостность, доступность) необходимо определить стационарные вероятности $\{\pi_i, i \in S\}$ для полумарковского процесса. Эти вероятности определяются в терминах стационарных вероятностей π_i^d и среднего времени пребывания h_i в состоянии для вложенного марковского процесса с дискретным временем. Записав матрицу P, которая описывает вероятности перехода этого марковского процесса, и решив матричное уравнение

$$\pi^d = \pi^d \times P,\ \sum_i \pi_i^d = 1,\ \ i \in S,$$

получаем стационарное распределение вероятностей для вложенного марковского процесса.

Далее вычисляется среднее время пребывания h_i для всех состояний на основании допущений о вероятностном распределении времени пребывания в каждом состоянии. Стационарные вероятности для полумарковского процесса определяются по формуле

$$\pi_i = \frac{\pi_i^d \cdot p_i}{\sum \pi_j^d \cdot p_j}.$$

Количественные значения атрибутов безопасности (конфиденциальность, доступность, целостность) определяются как вероятности того, что система владеет указанным атрибутом, т. е., например, доступность системы определяется как

$$A = 1 - \sum_{i \in SA} p_i,$$

где SA – множество состояний системы, в которых она является недоступной.

Аналогично можно определить и другие показатели, например вероятность успешной DoS-атаки, вероятность компрометации определенной информации и т. д. [38].

1.5. Анализ особенностей известных прикладных моделей безопасности

В данном пункте рассмотрим ряд прикладных моделей, которые используются на практике для построения и анализа кибернетических СЗИ.

Элементы и идеи данных моделей будут использованы в последующих разделах.

1.5.1. Поверхность атаки ИКС

Авторы работы [39] предлагают подход для сравнения защищенности однотипных ИКС, который базируется на измерении их поверхности атаки: чем больше поверхность атаки системы, тем менее защищенной она является. Оценка размера поверхности атаки выполняется в терминах ресурсов, которые используются при осуществлении атаки: предоставляемые наружу методы, каналы и данные. Поверхность атаки – это подмножество ресурсов системы, которые могут быть использованы злоумышленником для несанкционированного проникновения в систему и/или нарушения ее функционирования.

Рассматривается набор систем S, пользователь U и набор данных D. Для определенной системы $s \in S$ определяется ее окружение (среда) $E_s = \langle U, D, T \rangle$, где $T = S \setminus \{s\}$. Каждая система из S имеет набор методов. Метод системы получает входные аргументы и на выходе возвращает результат. Примером методов может быть публичное API для системы. Также каждая система имеет набор каналов передачи. Каналы системы *s* – это средства, с помощью которых пользователь U или другая система в среде могут взаимодействовать с *s*. Примерами каналов являются TCP/UDP сокеты, конечные точки удаленного вызова процедур, именованные каналы и т. д. Набор данных D состоит из элементов данных. Примерами элементов данных являются строки, URL-адреса, файлы, куки и пр. Пользователь U представляет злоумышленника, который атакует системы из S.

Для определения ресурсов, которые входят в поверхность атаки, авторы применяют подход входных и выходных точек. Входная точка – это метод,

который получает элементы данных непосредственно из окружения системы. Соответственно, выходная точка системы – это метод, посылающий элементы данных непосредственно в окружение системы.

Среди всех элементов данных в D выделяют элементы данных, которым система не может доверять, к ним принадлежат те элементы данных, которые система *s* непосредственно считывает из окружения или записывает в среду.

Набор входных и выходных точек системы *M*, каналов C и элементов данных, которым система не может доверять, *I* – это те ресурсы, которые злоумышленник может использовать для атаки, т. е. поверхность атаки для системы *s* – это тройка $\langle M, C, I \rangle$.

Кроме того, данная модель учитывает тот факт, что не все элементы поверхности атаки вносят одинаковый вклад в уменьшение защищенности системы. Например, для UNIX-систем метод m_1, который выполняется от имени обычного пользователя, делает значительно меньший вклад в поверхность атаки, чем метод m_2, который выполняется от имени суперпользователя. Для оценки вклада каждого ресурса в поверхность атаки авторы предлагают применять соотношение «эффективность – усилие» – отношение вреда, который злоумышленник может причинить системе, используя данный ресурс, к усилиям, затраченным на использование этого ресурса. Чем выше для ресурса соотношение «эффективность – усилие», тем больше его вклад в поверхность атаки.

Таким образом, процедура оценки поверхности атаки системы состоит из следующих шагов:

Шаг 1. Для заданной системы *s* и ее окружения E_s определяется множество входных и выходных точек *M*, множество каналов *C* и множество элементов данных *I*, которым система *s* не может доверять.

Шаг 2. Оценивается соотношение «эффективность – усилие»: $der_m(m)$, $\forall m \in M$, $der_m(m)$, $\forall m \in M$ и $der_c(c)$, $\forall c \in C$.

Шаг 3. Мерой поверхности атаки является тройка

$$\left\langle \sum_{m \in M} der_m(m), \sum_{c \in C} der_c(c), \sum_{d \in I} der_d(d) \right\rangle.$$

Авторы в работе [39] приводят ряд результатов, которые подтверждают эффективность данной методики и соответствие результатов ее применения предыдущим исследованиям и статистическим данным. Предложенный авто-

рами подход может быть применен как один из инструментов при проектировании и разработке защищенных ИКС для уменьшения рисков, связанных с кибербезопасностью. Стоит также отметить, что оценка поверхности атаки уже применяется как часть Microsoft Security Development Lifecycle.

1.5.2. Деревья атак

Дерево атак как метод для анализа надежности информационных и инженерных систем используется очень давно. В литературе он также фигурирует под названием «дерево ошибок», «дерево отказов», «дерево рисков» и т. п.

Деревья атак представляют собой методологию для описания угроз и способов противодействия для защиты системы. В развернутом виде схема позволяет наглядно представить безопасность системы и дает возможность просчитать защиту, сравнить методы защиты разных систем и пр.

Основная идея состоит в том, что с помощью дерева атак можно выполнять описание вариантов осуществления атаки для достижения определенной цели, которая стоит во главе дерева атак (является его вершиной) [40].

Каждый узел дерева представляет собой некоторую подцель, достижение которой при выполнении ряда условий позволяет злоумышленнику подняться по дереву на более высокий уровень. И так до тех пор, пока злоумышленник не достигнет вершины дерева (конечной цели) [41].

Пример деревьев атак приведен на рис. 1.4.

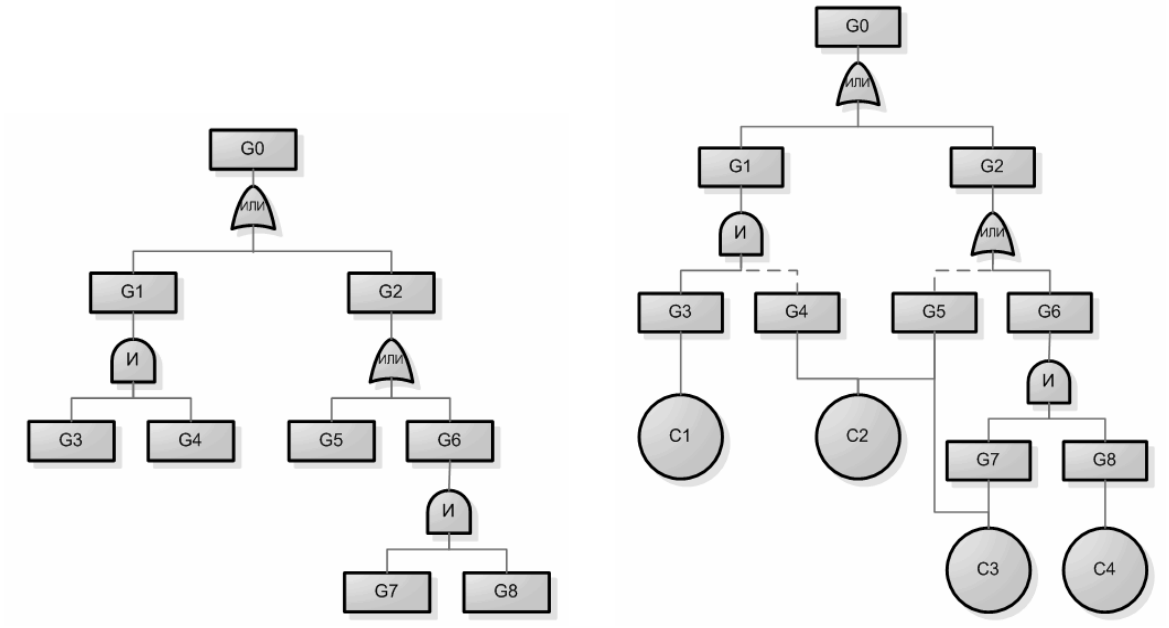


Рис. 1.4. Деревья атак

Если приписать каждому из узлов дерева некоторые значения, то это позволит выполнять некоторые расчеты. Такая схема называется «деревом И/ИЛИ».

Пользуясь деревьями атак, можно не только разрабатывать варианты кибератак и определять наиболее опасные сценарии, но и также разрабатывать контрмеры против них. Каждый из вариантов кибератаки должен быть закрыт контрмерой (на рис. 1.4 приведены контрмеры *C1*, ..., *C4*). Некоторые контрмеры могут закрывать сразу несколько вариантов (например, для подцели *G4* и *G5* – контрмера *C2*), а для некоторых вариантов может понадобиться несколько контрмер (для подцели *G5* – контрмеры *C2* и *C3*) [107].

Если модель системы защиты с полным перекрытием давала нам лишь перечень угроз и соответствующие механизмы защиты, которые их перекрывают, в виде списка, то с помощью дерева атак появляется возможность делать иерархическую декомпозицию, исследовать зависимости, анализировать разные сценарии атак и оптимально планировать размещение механизмов защиты.

Недостатком деревьев атак является то, что с их помощью нельзя описывать структуры графов, которые могут содержать циклы и повторяющиеся элементы.

1.5.3. Графы атак

Понятие графа атак появилось за счет проекции более общего понятия графа сценариев на область защиты информации. Граф атак – это такой граф сценариев, в котором каждый путь представляет атаку, то есть способ, которым злоумышленник может вмешаться в деятельность системы.

Существует много работ, посвященных построению и анализу СЗИ с использованием графов атак [42, 43, 44, 45, 46, 47]. Главное преимущество данного подхода состоит в том, что с помощью графа атак для действующей ИКС удается очень точно смоделировать возможные варианты проникновения злоумышленников в систему.

В работе [38] авторы рассматривают два алгоритма для генерации графов сценариев. Первый из них базируется на символической проверке модели, а результат его работы выражается в терминах логики дерева вычислений. Другой алгоритм использует анализ чистых состояний системы и результат его работы формулируется в терминах линейной темпоральной логики.

Также можно определить набор свойств, которые соответствуют защищенному состоянию системы, и для каждого из этих свойств построить граф

сценариев, приводящих к нарушению данных свойств, по одному из алгоритмов. Объединение этих графов сценариев и составит граф атак для системы.

Строить граф атак для системы можно вручную, но этот способ трудоемкий, и полученный граф может содержать ошибки, поэтому целесообразно применять автоматические способы построения, особенно если речь идет о кибератаках.

Граф кибератак может служить отправной точкой для анализа безопасности сети в разных направлениях: обнаружение вторжений, защита, расследование инцидентов и т. п. Графы кибератак могут быть использованы:

– для сбора информации о том, к каким атакам система является уязвимой и сколькими способами злоумышленник может достичь своей цели;

– принятия решений относительно предотвращения атак и выбора метрик безопасности.

При помощи построенного графа можно определить приоритетность закрытия той или иной уязвимости, визуально отобразить возможные пути атак или вычислить заданные метрики безопасности.

Авторы детально рассматривают два способа анализа графов атак: удаление одной вершины (действия) и минимизацию критического набора действий. Первый способ позволяет определить эффект от удаления отдельного действия из арсенала злоумышленника, второй – определяет набор действий, которые следует заблокировать, чтобы предотвратить достижение злоумышленником своей цели.

1.5.4. Топологический анализ уязвимостей

В рамках традиционного подхода к анализу защищенности ИКС каждая атака рассматривается отдельно. Это не соответствует истинному состоянию вещей, когда злоумышленник пытается комбинировать разные атаки для достижения своей цели. Для реалистического моделирования защищенной системы необходимо рассматривать взаимосвязи между разными атаками и уязвимостями. Топологический анализ уязвимостей, предложенный Р. Тростом [48], базируется на использовании графов атак и позволяет собрать информацию о защищенности сети с учетом разных путей проникновения злоумышленника в сеть.

Общая структура подхода приведена на рис. 1.5. Первым шагом является построение модели атаки на основании данных о конфигурации сети и

имеющихся уязвимостях. Данные о конфигурации сети могут включать отчеты сканера уязвимостей, результаты инвентаризации сети, правила, определенные на брандмауэре, и т. п. После этого строится граф атак, который учитывает как одно-, так и многошаговые атаки.

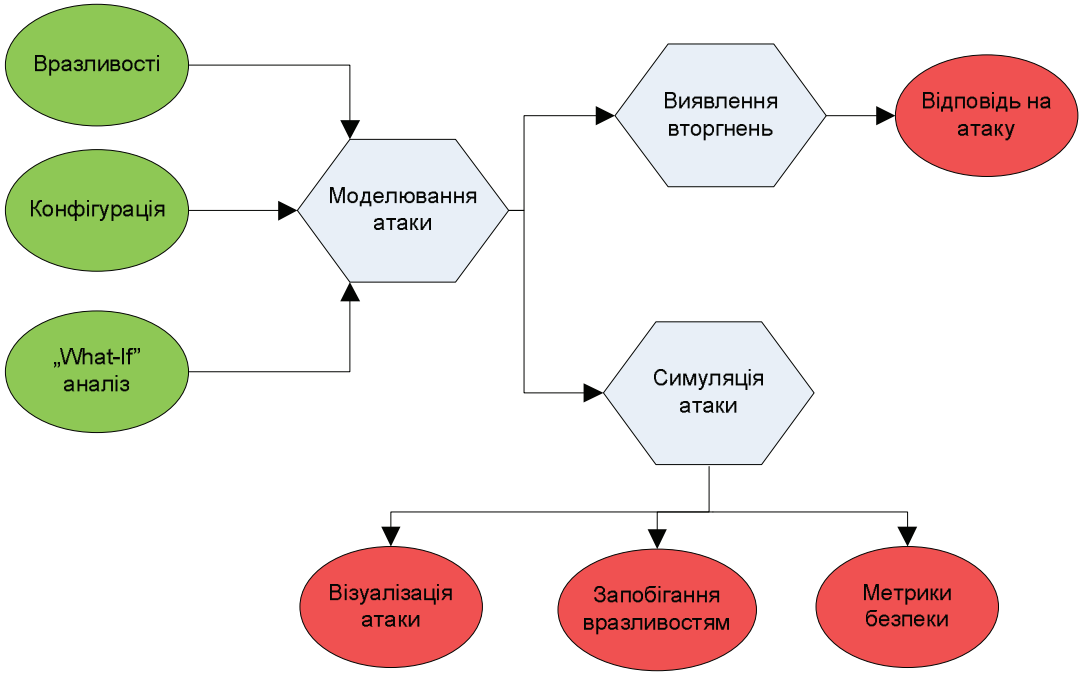


Рис. 1.5. Общая схема применения топологического анализа уязвимостей

Топологический анализ уязвимостей может быть применен для многих типов систем и атак – в том числе и не связанных с ИКС. Если же говорить о моделировании именно компьютерной сети, то авторы предлагают рассматривать сеть как совокупность хостов, разделенных на так называемые домены защиты (protection domains). Хосты, находящиеся в одном домене защиты, имеют неограниченный доступ к сервисам (в том числе и уязвимым) друг друга. Такой подход является альтернативой наличию полностью связанных подграфов в графе атак. Также домены могут быть вложенными.

Каждый хост сети включает ряд элементов и атрибутов, важных для моделирования сетевых атак. Примерами таких атрибутов могут быть операционная система, связи с уязвимыми сервисами на других хостах, хосты, которым данный хост доверяет, и т. д. Отдельными атрибутами могут быть средства защиты; в рамках данного подхода полагается, что эффективность средства защиты абсолютная, т. е. при его наличии для определенной уязвимости она полностью исключается из графа атак.

Как уже отмечалось, данные о структуре сети могут собираться автоматически, точно также данные об уязвимости можно получать из специализированных баз данных. Таким образом, человеку, выполняющему построение графа атак, остается задать только хосты источника угроз и критические хосты, на которые направлена атака. После этого программное средство автоматически строит граф атак для сети.

После генерации графа атак автоматически создается список компонентов сети, на которых следует установить дополнительные средства защиты для того, чтобы повысить защищенность целевых хостов. Одним из вариантов является усиление защиты непосредственно на источники атаки. Другой способ – усиление защиты критических ресурсов. Во всех случаях можно легко увидеть, сколько уязвимостей следует закрыть для того, чтобы достичь высшего уровня защищенности – исходя из этого, выбирается самый эффективный способ защиты. Например, для сети, изображенной на рис. 1.6, наиболее эффективным по количеству уязвимостей, которые нужно закрыть, будет третий вариант защиты.

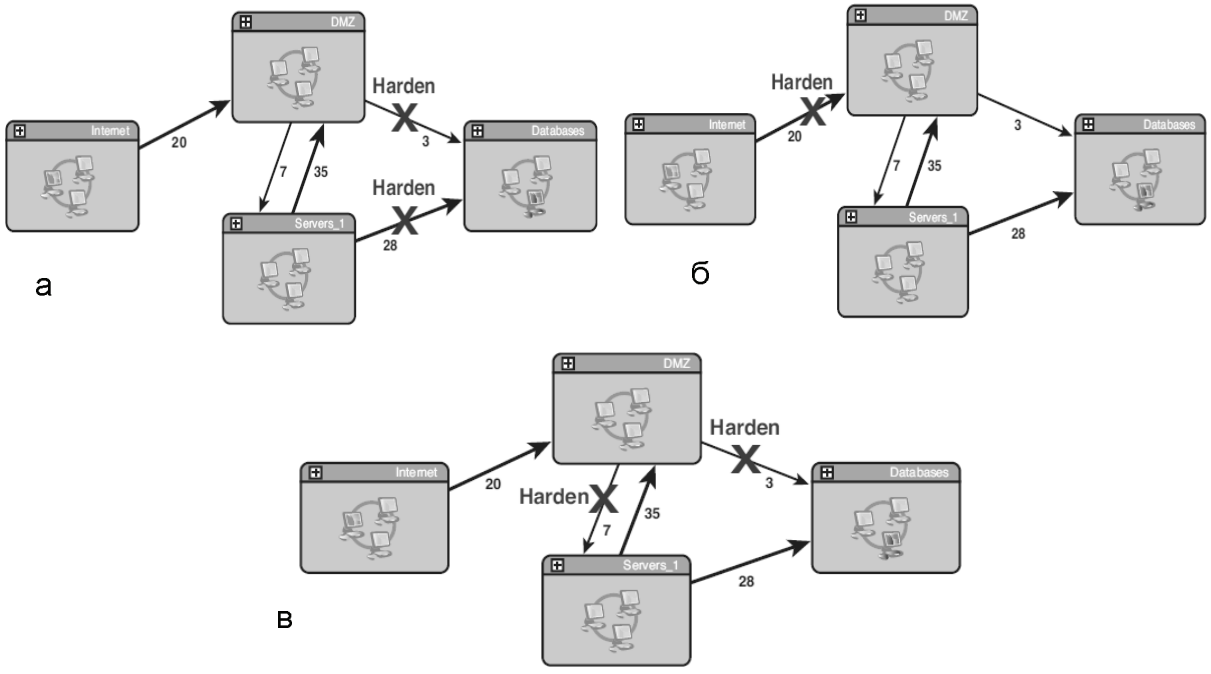


Рис. 1.6. Усиление защиты (*а*) цели атаки, (*б*) источника атаки, (*в*) промежуточных элементов – наиболее эффективный по количеству уязвимостей вариант

Данный подход, например, позволяет определить, где следует разместить сенсоры IDS, чтобы покрыть все пути атак минимальным количеством сенсоров. Также, зная пути атак, можно отфильтровывать ложные сообщения о вторжениях.

Построенный граф атак позволяет сопоставлять отдельные сигналы тревоги и рассматривать их как часть большой многошаговой атаки, т. е. применение топологического анализа уязвимостей может сделать реакцию на атаки более эффективной, одновременно минимизируя количество ложных тревог и стоимость системы защиты.

В то же время топологический анализ уязвимостей содержит средства только для моделирования/симуляции атак и, как любое средство моделирования, требует входных данных, поэтому эффективность такого моделирования напрямую зависит от эффективности существующих инструментов для сбора информации о конфигурации сети и имеющихся уязвимостей. Если входные данные будут неполными, то и полученный граф атак тоже будет содержать не все возможные пути атак.

Кроме того, при моделировании атак приходится ограничивать детальность этой модели, чтобы ее вычислительная сложность оставалась в разумных пределах.

1.6. Подходы к построению защищенных от киберугроз ИКС и анализ их защищенности

В предыдущих пунктах были рассмотрены как теоретические, так и прикладные модели, которые могут использоваться при построении защищенных от киберугроз ИКС. В данном пункте рассмотрим факторы, влияющие на уровень защищенности ИКС, последовательность разработки комплексной системы кибернетической защиты информации и способы оценки уровня результирующей защищенности.

1.6.1. Факторы, влияющие на уровень защищенности ИКС

Защищенная компьютерная система – компьютерная система, которая способна обеспечивать защиту обрабатываемой информации от определенных угроз. Этот термин чаще используется по отношению к вычислительным системам или их составляющим (программным продуктам, отдельным программно-аппаратным устройствам) [4].

Современная ИКС функционирует в киберпространстве и является структурно сложной системой, которая состоит из большого числа взаимосвязанных подсистем разного уровня сложности. Набор этих подсистем и их взаимодействие

между собой строится, исходя из тех целей, для которых создается ИКС. Соответственно, состав подсистем, настройка каждой из них, а также их взаимодействие будут влиять на защищенность всей системы в целом. Но, как известно, защищенность всей системы будет определяться наименее защищенным из ее компонентов.

Среди основных факторов, которые определяют защищенность системы, можно выделить следующие:

– качество программного и программно-аппаратного обеспечения всех уровней;

– архитектура системы;

– настройка параметров (конфигурация) программного и программно-аппаратного обеспечения;

– пользователи и киберпространство, в котором функционирует ИКС;

– организация и проверка уровня защиты информации в организации (комплексной системе кибернетической защиты информации).

В свою очередь, каждый из этих факторов может быть разбит на множество составляющих. Для примера качество и безопасность программного обеспечения, которое разрабатывается, будут определять методология разработки, квалификация команды разработчиков, используемая платформа и языки программирования, средства разработки и тестирования, а также многое другое.

Для каждого из уровней ИКС (рис. 1.7) существует большое количество материалов и рекомендаций относительно обеспечения их безопасности.

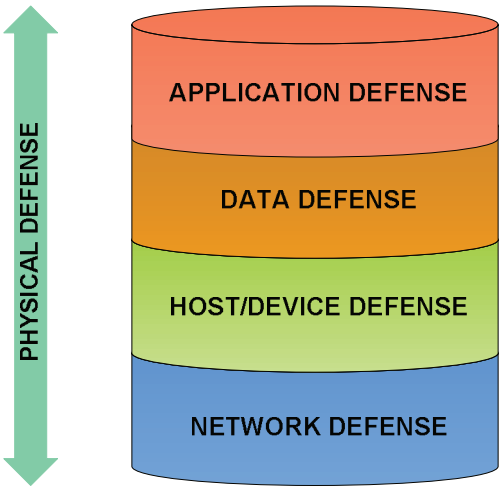


Рис. 1.7. Уровни защиты ИКС

Например, относительно безопасности разрабатываемых приложений существуют рекомендации по безопасному программированию для всех известных языков (C/C++, Java, C#, PHP и др.) [49, 50, 51, 52, 53].

Также есть ряд рекомендаций относительно организации самого процесса разработки программного обеспечения, направленных на уменьшение потенциального количества уязвимостей в итоговом продукте – Security Development Lifecycle (SDL) [54, 55, 56, 57, 58, 59], а также рекомендации по проведению тестирования безопасности программного обеспечения (fuzzing, peniteration testing) [60, 61, 62, 63, 64]. Если перейти на уровень использования программного и программно-аппаратного обеспечения, то разработчики выпускают рекомендации по безопасной настройке параметров своих продуктов [65, 66, 67, 68].

Также существуют рекомендации по организации архитектуры корпоративных сетей, их разбивке на сегменты, выделению зон с разными уровнями безопасности [69, 70, 71 ,72 ,73]. Но тут уже происходит качественный переход, когда не всегда можно однозначно указать, какой из языков программирования и какую функцию лучше использовать или в какое значение необходимо установить некоторый параметр. Данные рекомендации носят более общий характер и требуют дальнейшей интерпретации для непосредственного воплощения в настройках параметров конкретного программного или программно-аппаратного средства.

Исходя из этого, защита компьютерной системы от киберугроз в целом является сложным заданием, потому что необходимо принимать во внимание все рекомендации и факторы в совокупности, применять специальные способы защиты и при этом обеспечивать выполнение ИКС ее непосредственных бизнес-функций. Это все должно быть учтено при разработке КС КЗИ.

1.6.2. Разработка комплексной системы кибернетической защиты информации

Как уже упоминалось, для защиты ИКС от разного вида киберугроз создается КС КЗИ.

Для обеспечения методологического подхода к построению системы кибернетической защиты информации в компьютерных системах могут использоваться международные (ISO) [74, 75] и отечественные стандарты (ДСТУ) [76, 77, 78], а также НД ТЗИ [23, 9, 79, 80].

В своем большинстве подобные методологические подходы представляют собой структурированное описание последовательных задач и документов, которые необходимо подготовить для построения системы защиты; они нацелены на дальнейшую процедуру международной или государственной сертификации или аттестации.

Но, как уже говорилось, даже четкое следование этим стандартам и нормативным документам не будет гарантировать отсутствие проблем во время построения системы кибернетической защиты информации.

В соответствии с НД ТЗИ [81] порядок создания КС КЗИ в ИКС является независимым от того фактора, находится ли данная система на стадии проектирования, эксплуатации либо модернизации. Но, исходя из практического опыта авторов и концепции глубокой эшелонированной защиты (defense-in-depth) [82], необходимо, чтобы КС КЗИ начинала создаваться на стадии формирования требований к ИКС, во время ее проектирования. Откладывание создания системы защиты на более поздние этапы может привести к дополнительным расходам и менее эффективным решениям, и, возможно, в худшем случае, к невозможности достичь необходимого уровня защищенности (безопасности) [83].

Исходя из этого, начинать проектировать КС КЗИ (или хотя бы задумываться о ней) есть смысл как можно раньше, а также обязательно интегрировать ее разработку в общий процесс проектирования ИКС, как это предлагается в методологии Security Development Lifecicle (SDL) [56].

Но для начала проектирования КС КЗИ необходим набор входных данных и условий, которые будут появляться как результаты соответствующих этапов проектирования самой ИКС. Кроме того, ряд решений, принятых при проектировании КС КЗИ, также будут влиять на проектируемую архитектуру ИКС, приводя к необходимости пересмотра ее структуры, архитектуры приложений, сегментации сети и топологии.

Для того чтобы определить, какие проблемы могут возникать во время построения системы защиты с использованием методологических подходов, рассмотрим, для примера, процесс создания КС КЗИ, который приводится в отечественных нормативных документах технической защиты информации, а в конце раздела приведем перечень проблем, которые возникают во время следования нормативным документам.

Исходя из нормативных документов, КС КЗИ – это совокупность организационных, инженерных мероприятий, программно-аппаратных средств, которые обеспечивают защиту информации в ИКС.

В этом определении под защитой информации в ИКС подразумевают деятельность, направленную на обеспечение безопасности информации (информационную безопасность – ИБ), обрабатываемой в ИКС, и ИКС в целом, которая позволяет избежать или затруднить возможность реализации угроз, а также снизить величину потенциального убытка в результате реализации угроз.

Приведем определения понятий, которыми оперирует данное определение: «безопасность информации» и «угрозы».

Безопасность информации – это состояние информации, в котором обеспечивается сохранение определенных политикой безопасности свойств информации. Многолетний опыт защиты информации в ИКС позволил определить главные свойства информации, сохранение которых позволяет гарантировать сохранение ценности информационных ресурсов:

– конфиденциальность;

– целостность;

– доступность.

Следующим основным фундаментальным понятием является «политика безопасности» – это совокупность законов, правил, ограничений, рекомендаций, инструкций и т. д., которые регламентируют порядок обработки информации в ИКС. Таким образом, именно политика безопасности обуславливает употребление тех или иных мероприятий защиты, которые позволяют поддерживать безопасность информации.

Теперь определим термин «угроза» – это любые обстоятельства или события, которые могут быть причиной нарушения политики безопасности и/или нанесения убытка ИКС. Связь между этими понятиями изображена на диаграмме (рис. 1.8).

Разработка КС КЗИ представляет собой итеративный процесс, на каждом этапе которого происходит уточнение структуры и состава объекта защиты, возможных угроз, политики безопасности, механизмов безопасности и технических средств защиты.

В нормативных документах прописан следующий порядок создания КС КЗИ.

Рис. 1.8. Диаграмма связей между понятиями в ИБ

Создание КС КЗИ должно начинаться с анализа объекта защиты и возможных угроз. Прежде всего должны быть определены ресурсы АС, которые нуждаются в защите. Угрозы должны быть определены в терминах вероятности их реализации и величины возможных убытков. На основе анализа угроз, существующих в системе уязвимостей, эффективности уже реализованных мер защиты для всех ресурсов, которые подлежат защите, должны быть оценены риски. Риск представляет собой функцию вероятности реализации определенной угрозы, вида и величины причиненного ущерба. Величина риска может быть выражена в денежном эквиваленте или в виде формальной оценки (высокий, низкий и т. д.). На основании выполненной работы должны быть выработаны мероприятия защиты, воплощение которых в жизнь позволило бы снизить уровень окончательного риска до приемлемого уровня. Итогом данного этапа работ должна стать сформулированная или скорректированная политика безопасности.

На основании проведенного анализа рисков и сформулированной политики безопасности разрабатывается план защиты, который включает в себя описание последовательности и содержание всех стадий и этапов жизненного

цикла КС КЗИ, которые должны соответствовать стадиям и этапам жизненного цикла АС. Стоимость мероприятий по защите информации должна быть адекватна размеру возможных убытков [22].

Из приведенного короткого описания порядка разработки КС КЗИ выделим следующие задачи, которые необходимо выполнить во время создания системы защиты [81]:

– разработка формальной политики безопасности;

– определение всех угроз;

– определение вероятности реализации угроз и анализ рисков;

– выбор адекватных механизмов защиты и улучшение защищенности ИКС;

– оценка защищенности ИКС и выявление ее слабых компонент.

Именно при выполнении данных задач и будет возникать большинство проблем. Проблемы будут заключаться в том, что качество и правильность полученных результатов будут определяться субъективным фактором и зависеть от профессиональности проектировщиков. Кроме того, данные задачи могут иметь неоднозначность решения и допускать несколько различных интерпретаций.

Иногда в задачах ИБ удается достичь оптимального решения в общепринятом понимании и доказать его существование. Однако, когда решение многоальтернативное, найти оптимальное решение удается только в частных случаях [84]. Кроме того, оптимальное решение может не быть таковым с точки зрения функциональности системы, так как система защиты не является самоцелью и должна выполнять второстепенную функцию по сравнению с основными бизнес-функциями системы. Причины подобных проблем вытекают из сути самой теории защиты информации.

1.6.3. Оценка защищенности ИКС

После построения КС КЗИ необходимо выполнить оценку полученного уровня защищенности ИКС.

На сегодняшний день не существует каких-либо стандартизированных методик анализа защищенности ИКС, поэтому действия экспертов или аудиторов могут существенно отличаться в зависимости от ситуации [85].

Проанализировав ряд источников [86, 87, 88, 89], приведем следующие подходы по оценке уровня защищенности:

– аудит (Audit);

– проверка соответствия (Compliance);

– тестирование на проникновение (Peniteration testing).

Цели, которые ставятся перед этими подходами, приведены на диаграмме на рис. 1.9.

Рис. 1.9. Варианты оценки уровня защищенности ИКС

Выводы по результатам проверок могут быть также разделены на категории:

– относительно архитектуры, настроек и конфигураций, правильности внедрения политики безопасности;

– относительно соответствия корпоративным, международным стандартам и нормативным документам;

– относительно уязвимостей в системе и возможности проникновения.

Некоторые из них непосредственно влияют на безопасность ИКС, например, уязвимости в системе и возможность проникновения; а некоторые лишь опосредованно, например, соответствие стандартам (т. е. система может быть защищенной, но не соответствовать стандартам).

Следует также отметить, что данные результаты проверок не дают интегрального представления об уровне кибербезопасности системы, по которому можно определять, например, насколько система сегодня является более/менее защищенной, чем вчера.

Кроме этого, оценка защищенности делается еще и для того, чтобы дать рекомендации для устранения ошибок, а также повышения уровня защищенности системы в целом, и после реализации предоставленных рекомендаций повторная проверка должна показать улучшение состояния защиты.

Из этого следует, что, кроме описательных (неформальных) критериев, должны быть разработаны подходы или методики, по которым можно было бы интегрально определить уровень или степень кибербезопасности системы, а также определить эффективность и влияние внедряемых мероприятий и рекомендаций.

1.6.4. Метрики для оценки уровня защищенности ИКС

Одним из средств формальной оценки безопасности могут служить метрики. Рассмотрим более детально их преимущества, недостатки и виды метрик безопасности. По сути, метрики – это единственное более-менее формализованное средство для оценки уровня защищенности ИКС.

Метрики сами по себе являются не целью, а важной частью системы управления, с помощью которой можно направлять и контролировать работу IT-подразделения [90].

Основными заданиями, которые ставятся при выборе метрик, являются следующие [91]:

– помочь проанализировать некоторые характеристики предметной области или определять их состояние;

– получить количественные показатели выбранной предметной области;

– способствовать получению ответов на вопросы «до и после», «что, если», «почему, почему нет».

Не существует однозначного определения понятия «метрики безопасности», поэтому приведем несколько [92]:

«Метрика безопасности – способ применения количественного, статистического и/или математического анализа для измерения «безопасных» стоимости, преимуществ, удач, неудач, тенденций и нагрузок»;

«На высоком уровне метриками являются численные измерения некоторых аспектов системы или предприятия. Для безопасности информационной системы измерения должны быть связаны с аспектами системы, которые вносят вклад в ее безопасность. Для объектов (система, продукт и др.), для

которых безопасность является значимой, можно выделить набор атрибутов, которые в совокупности будут характеризовать защищенность этого объекта. Кроме этого, метрики безопасности (или комбинация метрик безопасности) являются количественной мерой атрибута, которыми данный объект обладает. Метрики безопасности могут строиться на основе измерений более низких уровней».

Основные принципы, по которым следует подходить к выбору метрик, следующие [91]:

– воспроизводимость и независимость – однозначно измеряться, без влияния «экспертного мнения»;

– легкость измерения – доступность для расчетов и анализа, желательно автоматически;

– наличие количественного или процентного показателя (а не «высокий», «средний», «низкий»), который выражается в некоторых единицах («ошибки», «часы», «стоимость», …);

– актуальность для принятия решений.

Существует большое количество возможных метрик безопасности – от простых до тех, которые интегрируют в себе несколько разных показателей. Приведем некоторые из них [93, 94, 39]:

– выявление вирусов в файлах пользователей и электронной почте;

– попытки неудачных вводов паролей при входе в систему;

– попытка проникновения/атаки;

– доступ к опасным сайтам;

– неудачное имя при входе в систему;

– внутренняя попытка НСД;

– стойкость паролей;

– время, необходимое для установки обновлений;

– процент проведенных проверок и аудитов;

– число инцидентов, связанных с информационной безопасностью.

Исходя из текущего состояния метрик безопасности, можно сделать вывод, что они почти все базируются на определенных показателях уже работающей системы, выбранных субъективным образом, и не всегда могут отображать реальное состояние информационной безопасности в ИКС, а также удовлетворять приведенным выше требованиям к метрике.

Также текущие метрики безопасности не могут быть использованы во время проектирования и разработки комплексной системы защиты ИКС. Единственно, что статистические данные, связанные с показателями определенного объекта, продукта или классом системы, могут служить входными данными для проектирования и моделирования разрабатываемой системы.

Среди дальнейших направлений развития метрик безопасности предлагается следующее [92]:

– определить «хорошие» оценки для системы безопасности;

– уменьшить человеческое влияние на измерения и их последствия;

– предложить более систематические и быстрые подходы для получения соответствующих показателей;

– понять состав и зависимости механизмов безопасности.

Для достижения этого задания исследования должны быть направлены на разработку формальных моделей с уровнем детализации, достаточным для моделирования поведения системы и текущего определения ее уровня защищенности.

1.7. Варианты постановок задач синтеза систем кибернетической защиты

В завершающем пункте данной главы рассмотрим ряд актуальных задач построения защищенных ИКС [95, 96, 97, 98].

Среди задач синтеза систем кибернетической защиты информации с оптимальными свойствами можно выделить следующие: структурный синтез системы кибернетической защиты информации с оптимальным уровнем защищенности информации, синтез параметров системы кибернетической защиты информации с оптимальным уровнем защищенности информации. В этих задачах в критериях качества используются технологические показатели, а в качестве ограничений рассматриваются экономические показатели.

Актуальным является класс задач, обратных к вышеприведенным, где критерием качества является экономический показатель, а в ограничениях – технологические показатели – это структурный синтез системы кибернетической защиты информации с оптимальными экономическими показателями, синтез параметров системы кибернетической защиты информации с оптимальными экономическими показателями.

Перечень данных задач и их комбинации в соответствии с [99] приведены в табл. 1.2.

Таблица 1.2

Основные задачи синтеза систем защиты информации с оптимальными свойствами

Классы задач оптимального синтеза	**Критерий**	**Ограничения**	**Метод оптимизации**
Синтез структуры системы кибернетической защиты информации	Максимизация технологического показателя уровня защищенности	Фиксированная стоимость системы кибернетической защиты информации	Линейное булевое программирование, сепарабельное программирование
Синтез параметров механизмов защиты информации	Максимизация технологического показателя уровня защищенности	Фиксированная стоимость системы кибернетической защиты информации	Градиентные методы
Синтез структуры системы кибернетической защиты информации	Минимизация стоимости системы защиты	Фиксированный уровень защищенности информации	Линейное булевое программирование, сепарабельное программирование
Синтез параметров механизмов защиты информации	Минимизация стоимости системы защиты	Фиксированный уровень защищенности информации	Градиентные методы
Комбинированные задачи структурного и параметрического синтеза	Комбинированный	Комбинированные	Комбинированные

В зависимости от структуры математических постановок задач оптимизации поиск экстремумов осуществляется с использованием разных методов математического программирования – методов ЛБП, методов сепарабельного программирования, градиентных методов и др.

Запишем задачи оптимального синтеза систем защиты, которые приведены в табл. 1.2, в формальном виде.

При построении системы защиты будем исходить из модели безопасности с полным перекрытием (модель Клементса–Хофмана). Систему защиты можно представить как набор механизмов защиты M, которые характеризуются некоторым параметром эффективности (прочности) K и имеют соответствующую стоимость C_M.

Тогда приведенные задачи оптимального синтеза будут иметь следующий вид:

1. Синтез структуры системы кибернетической защиты информации с оптимальным уровнем защищенности и фиксированной стоимостью системы защиты $C_{\max}$:

$$\begin{cases} F(M,K) \to \max\limits_{M} \\ F_1(M,K) \le C_{\max} \end{cases}.$$

Здесь и далее $F(M,K)$ – некоторый функционал, который будет характеризовать защищенность системы от кибератак и зависеть от выбранной структуры механизмов защиты M и их параметров эффективности K. В качестве характеристики защищенности системы при этом могут выступать один из видов риска, вероятность успешности кибератаки, количество уязвимых мест в системе, некоторый технологический показатель и т. п., а $F_1(M,K)$ – некоторый функционал, который будет характеризовать стоимостную (экономическую) составляющую системы защиты. В простейшем случае функционал будет определяться как суммарная стоимость выбранных механизмов защиты $F_1(M,K) = \sum\limits_i C_{M_i}$.

2. Синтез параметров системы кибернетической защиты информации с оптимальным уровнем защищенности информации и фиксированной стоимостью системы защиты:

$$\begin{cases} F(M,K) \to \max\limits_{K} \\ F_1(M,K) \le C_{\max} \end{cases}.$$

Основным отличием межу параметрическим и структурным синтезом с точки зрения математических методов, которые используются во время оптимизации, есть то, что в первом случае параметры являются непрерывными переменными, в то время как во втором – механизмы защиты являются дискретными переменными (булевыми). От этого будет зависеть, какие методы математического программирования будут применяться.

3. Синтез структуры системы кибернетической защиты информации с минимальной стоимостью системы защиты и фиксированным уровнем защищенности информации [100]:

$$\begin{cases} F(M,K) \geq F^* \\ F_1(M,K) \to \min\limits_{M} \end{cases}.$$

4. Синтез параметров системы кибернетической защиты информации с минимальной стоимостью системы защиты и фиксированным уровнем защищенности информации:

$$\begin{cases} F(M,K) \geq F^* \\ F_1(M,K) \to \min\limits_{K} \end{cases}.$$

Для решения приведенных задач в первую очередь необходимо формализовать и получить в явном виде функционалы $F(M,K)$ и $F_1(M,K)$. После этого, используя соответствующие методы оптимизации математического программирования, можно будет найти оптимальные параметры для построения систем кибернетической защиты.

Построению нескольких из возможных вариантов функций $F(M,K)$ и $F_1(M,K)$ будут посвящены следующие главы.

Выводы

В данной главе было дано определение ИКС, ее основные характеристики, а также факторы, которые влияют на уровень ее защищенности.

Были рассмотрены базовые теоретические и прикладные модели безопасности из теории защиты информации и проанализированы способы для оценки уровня защищенности ИКС.

В конце главы были приведены различные варианты постановки задачи синтеза систем кибернетической защиты информации и сделан вывод о необходимости наличия функции для оценки уровня защищенности от кибератак в явном виде.

2. ОСОБЕННОСТИ ЗАДАЧИ ПОСТРОЕНИЯ ЗАЩИЩЕННОЙ ИКС С ОТКРЫТОЙ АРХИТЕКТУРОЙ

В данной главе исследуем условия, особенности и закономерности функционирования ИКС с открытой архитектурой с точки зрения обеспечения ее кибернетической защиты.

Определим, с учетом условий и закономерностей функционирования ИКС, моделируемую характеристику, структуру и набор параметров разрабатываемой математической модели. Обоснуем выбор множества объектов моделируемой системы, а также способ представления в разрабатываемой модели множества кибернетических угроз, реализуемых по отношению к данным объектам.

2.1. Исследование закономерностей функционирования ИКС с открытой архитектурой

Как показано в п. 1.2, реализация кибернетических угроз и функционирование механизмов защиты в ИКС с открытой архитектурой имеют ряд особенностей, которые должны быть учтены в структуре разрабатываемых моделей. Для исследования влияния этих особенностей воспользуемся логическими моделями функционирования таких ИКС, построенными с использованием предложенного в работе [101] аппарата схем функциональной целостности (СФЦ), позволяющего графически представлять функционально полный набор логических операций «И», «ИЛИ», «НЕ».

На рис. 2.1 приведена построенная с использованием аппарата СФЦ граф-модель процесса обработки информации в гипотетической ИКС с двухуровневым стеком протоколов.

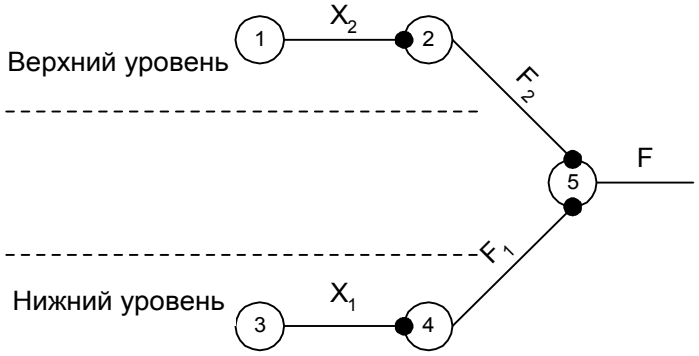


Рис. 2.1. Граф-модель процесса обработки информации в ИКС с двухуровневым стеком протоколов

Система логических уравнений, описывающих данную модель, имеет следующий вид:

$$\begin{cases} F_2 = X_2 \\ F_1 = X_1 \\ F = F_1 \cdot F_2 = X_1 \cdot X_2 \end{cases}, \qquad (2.1)$$

где F_2 – логическая функция, описывающая условия работоспособности средств ИКС, реализующих протокол верхнего уровня;

X_2 – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол верхнего уровня;

F_1 – логическая функция, описывающая условия работоспособности средств ИКС, реализующих протокол нижнего уровня;

X_1 – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол нижнего уровня;

F – логическая функция, описывающая условия работоспособности ИКС с двухуровневым стеком протоколов.

Приведенный на рис. 2.1 граф имеет монотонную структуру, т. е. из каждой начальной вершины существует только один путь в конечную. Это подтверждается и видом логической функции F, не содержащей операций дизъюнкции. С учетом структуры графа полученные результаты могут быть легко обобщены на случай системы с N уровнями стека протоколов:

$$F_N = \bigwedge_{i=1}^{N} X_i, \qquad (2.2)$$

где F_N – логическая функция, описывающая условия работоспособности ИКС с N-уровневым стеком протоколов;

X_i – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол i-го уровня.

Для дальнейшего анализа включим в модель системы переменные, отражающие факты наличия/отсутствия кибернетических угроз, реализуемых на различных уровнях стека протоколов. Соответствующая граф-модель приведена на рис. 2.2.

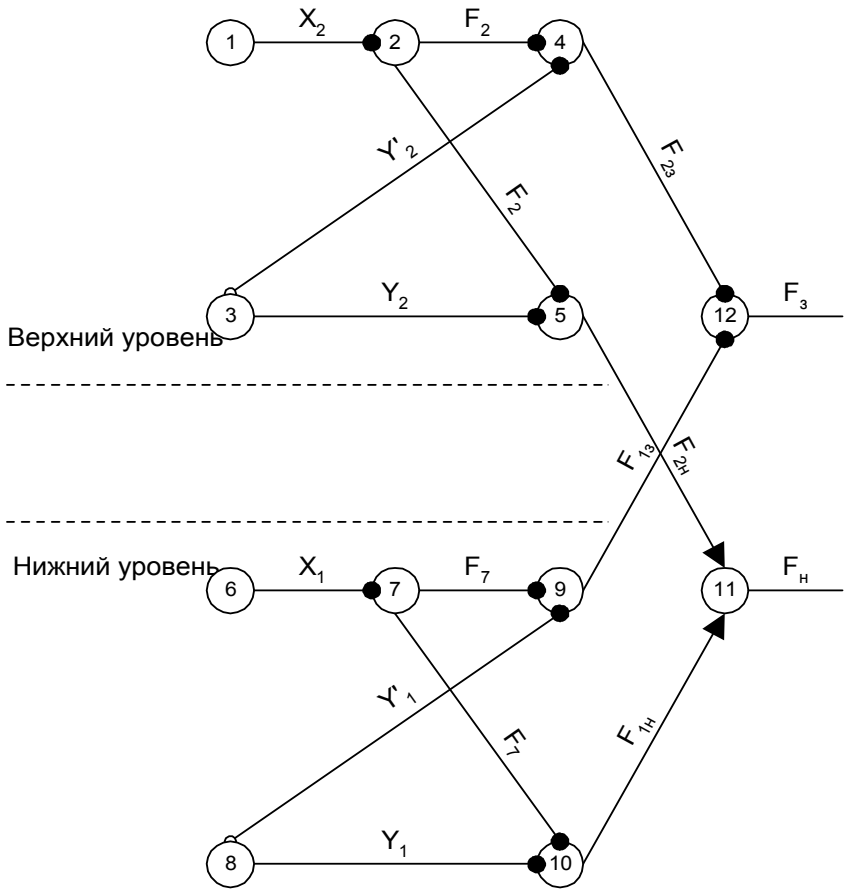


Рис. 2.2. Граф-модель процесса обработки информации
с учетом кибернетических угроз

Данная модель описывается следующей системой логических уравнений:

$$\begin{cases} F_2 = X_2 \\ F_{2з} = F_2 \cdot Y'_2 = X_2 \cdot Y'_2 \\ F_{2н} = F_2 \cdot Y_2 = X_2 \cdot Y_2 \\ F_7 = X_1 \\ F_{1з} = F_7 \cdot Y'_1 = X_1 \cdot Y'_1 \\ F_{1н} = F_7 \cdot Y_1 = X_1 \cdot Y_1 \\ F_з = F_{1з} \cdot F_{2з} \\ F_н = F_{1н} \vee F_{2н} \end{cases} . \qquad (2.3)$$

Полученные после выполнения соответствующих преобразований выражения для логических функций, описывающих условия сохранения и условия нарушения кибернетической безопасности, имеют вид

$$F_з = F_{1з} \cdot F_{2з} = X_1 \cdot Y'_1 \cdot X_2 \cdot Y'_2 ; \qquad (2.4)$$

$$F_н = F_{1н} \vee F_{2н} = X_1 \cdot Y_1 \vee X_2 \cdot Y_2 , \qquad (2.5)$$

где $F_з$ – логическая функция, описывающая условия сохранения кибернетической безопасности ИКС с двухуровневым стеком протоколов;

$F_{1з}$ – логическая функция, описывающая условия сохранения кибернетической безопасности средств ИКС, реализующих протокол нижнего уровня;

$F_{2з}$ – логическая функция, описывающая условия сохранения кибернетической безопасности средств ИКС, реализующих протокол верхнего уровня;

X_1 – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол нижнего уровня;

Y_1 – логическая переменная (Y'_1 – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на нижнем уровне стека протоколов;

X_2 – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол верхнего уровня;

Y_2 – логическая переменная (Y'_2 – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на верхнем уровне стека протоколов;

$F_н$ – логическая функция, описывающая условия нарушения кибернетической безопасности ИКС с двухуровневым стеком протоколов;

$F_{1н}$ – логическая функция, описывающая условия нарушения кибернетической безопасности средств ИКС, реализующих протокол нижнего уровня;

$F_{2н}$ – логическая функция, описывающая условия нарушения кибернетической безопасности средств ИКС, реализующих протокол верхнего уровня;

Приведенный на рис. 2.2 граф имеет монотонную структуру, однако логическая функция $F_н$ содержит операцию дизъюнкции, а это означает, что в ИКС имеются альтернативные пути развития опасного состояния системы (ОСС) при реализации кибернетических угроз на различных уровнях стека протоколов. На случай системы с N уровнями стека протоколов полученные выражения обобщаются следующим образом:

$$F_{Nз} = \bigwedge_{i=1}^{N} X_i \cdot Y'_i; \tag{2.6}$$

$$F_{Nн} = \bigvee_{i=1}^{N} X_i \cdot Y_i, \tag{2.7}$$

где $F_{Nз}$ – логическая функция, описывающая условия сохранения кибернетической безопасности ИКС с N-уровневым стеком протоколов;

X_i – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол i-го уровня;

Y_i – логическая переменная (Y'_i – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на протоколе *i*-го уровня;

$F_{Nн}$ – логическая функция, описывающая условия нарушения кибернетической безопасности ИКС с *N*-уровневым стеком протоколов.

Далее включим в модель системы переменные, отражающие факты наличия/отсутствия механизмов защиты, функционирующих на различных уровнях стека протоколов. Соответствующая граф-модель приведена на рис. 2.3.

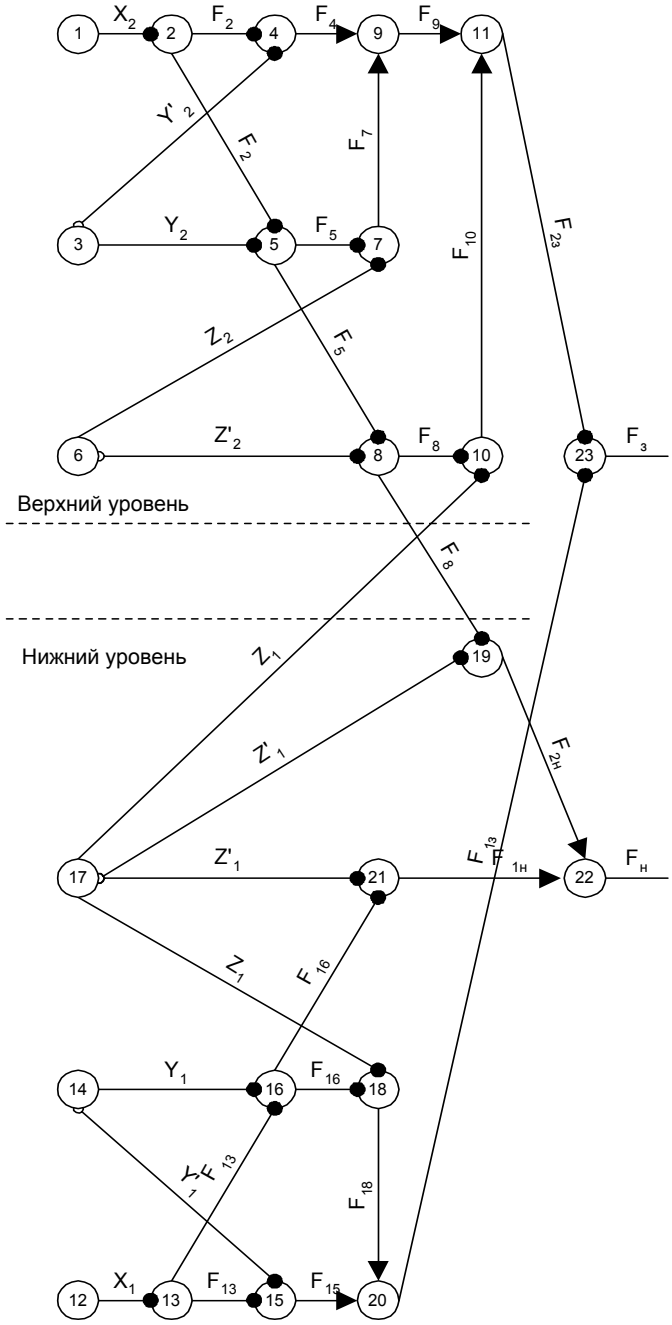


Рис. 2.3. Граф-модель процесса обработки информации
с учетом кибернетических угроз и механизмов защиты

Данная модель описывается следующей системой логических уравнений:

$$
\begin{cases}
F_2 = X_2 \\
F_4 = F_2 \cdot Y'_2 = X_2 \cdot Y'_2 \\
F_5 = F_2 \cdot Y_2 = X_2 \cdot Y_2 \\
F_7 = F_5 \cdot Z_2 = X_2 \cdot Y_2 \cdot Z_2 \\
F_8 = F_5 \cdot Z'_2 = X_2 \cdot Y_2 \cdot Z'_2 \\
F_9 = F_4 \vee F_7 = X_2 \cdot Y'_2 \vee X_2 \cdot Y_2 \cdot Z_2 \\
F_{10} = F_8 \cdot Z_1 = Z_1 \cdot X_2 \cdot Y_2 \cdot Z'_2 \\
F_{2з} = F_9 \vee F_{10} = X_2 \cdot Y'_2 \vee X_2 \cdot Y_2 \cdot Z_2 \vee Z_1 \cdot X_2 \cdot Y_2 \cdot Z'_2 \\
F_{2н} = F_8 \cdot Z'_1 = Z'_1 \cdot X_2 \cdot Y_2 \cdot Z'_2 \\
F_{13} = X_1 \\
F_{15} = F_{13} \cdot Y'_1 = X_1 \cdot Y'_1 \\
F_{16} = F_{13} \cdot Y_1 = X_1 \cdot Y_1 \\
F_{18} = F_{16} \cdot Z_1 = X_1 \cdot Y_1 \cdot Z_1 \\
F_{1з} = F_{15} \vee F_{18} = X_1 \cdot Y'_1 \vee X_1 \cdot Y_1 \cdot Z_1 \\
F_{1н} = F_{16} \cdot Z'_1 = X_1 \cdot Y_1 \cdot Z'_1 \\
F_з = F_{1з} \cdot F_{2з} = (X_1 \cdot Y'_1 \vee X_1 \cdot Y_1 \cdot Z_1) \cdot (X_2 \cdot Y'_2 \vee X_2 \cdot Y_2 \cdot Z_2 \vee \\
\qquad\qquad \vee Z_1 \cdot X_2 \cdot Y_2 \cdot Z'_2) \\
F_н = F_{1н} \vee F_{2н} = X_1 \cdot Y_1 \cdot Z'_1 \vee X_2 \cdot Y_2 \cdot Z'_1 \cdot Z'_2
\end{cases}
\qquad (2.8)
$$

Окончательные выражения для логических функций, описывающих условия предотвращения кибернетических угроз (условия сохранения кибернетической безопасности) и условия нарушения кибернетической безопасности имеют вид:

$$
\begin{aligned}
F_з = F_{1з} \cdot F_{2з} = X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee X_1 \cdot X_2 \cdot Y_1 \cdot Z_1 \vee \\
\vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z_1 \cdot Z'_2 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z_2
\end{aligned}
\qquad (2.9)
$$

$$
F_н = F_{1н} \vee F_{2н} = X_1 \cdot Y_1 \cdot Z'_1 \vee X_2 \cdot Y_2 \cdot Z'_1 \cdot Z'_2, \qquad (2.10)
$$

где $F_з$ – логическая функция, описывающая условия сохранения кибернетической безопасности ИКС с двухуровневым стеком протоколов;

$F_{1з}$ – логическая функция, описывающая условия сохранения кибернетической безопасности средств ИКС, реализующих протокол нижнего уровня;

$F_{2з}$ – логическая функция, описывающая условия сохранения кибернетической безопасности средств ИКС, реализующих протокол верхнего уровня;

X_1 – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол нижнего уровня;

X_2 – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол верхнего уровня;

Y_1 – логическая переменная (Y'_1 – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на нижнем уровне стека протоколов;

Y_2 – логическая переменная (Y'_2 – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на верхнем уровне стека протоколов;

Z_1 – логическая переменная (Z'_1 – ее инверсия), характеризующая факт работоспособности механизма защиты информации на нижнем уровне стека протоколов;

Z_2 – логическая переменная (Z'_2 – ее инверсия), характеризующая факт работоспособности механизма защиты информации на верхнем уровне стека протоколов;

$F_н$ – логическая функция, описывающая условия нарушения кибернетической безопасности ИКС с двухуровневым стеком протоколов;

$F_{1н}$ – логическая функция, описывающая условия нарушения кибернетической безопасности средств ИКС, реализующих протокол нижнего уровня;

$F_{2н}$ – логическая функция, описывающая условия нарушения кибернетической безопасности средств ИКС, реализующих протокол верхнего уровня.

В отличие от приведенных на рис. 2.1 и 2.2, полученный граф имеет немонотонную структуру, а логические функции $F_з$ и $F_н$ содержат операции дизъюнкции, т. е. в ИКС имеются как альтернативные пути развития ОСС при реализации кибернетических угроз на различных уровнях стека протоколов, так и альтернативные пути предотвращения кибернетических угроз механизмами защиты.

На случай системы с N уровнями стека протоколов полученные выражения можно обобщить следующим образом:

$$F_{Nз} = \bigwedge_{i=1}^{N} X_i \cdot \Big(\bigwedge_{i=1}^{N} Y'_i \vee \Big(\bigvee_{i=1}^{N}\Big(Y_i \cdot \bigwedge_{j=1}^{i-1} Y'_j \cdot \Big(\bigvee_{j=1}^{i} Z_j \cdot \bigwedge_{k=j+1}^{i} Z'_k\Big)\Big)\Big)\Big); \qquad (2.11)$$

$$F_{Nн} = \bigvee_{i=1}^{N} (X_i \cdot Y_i \cdot \bigwedge_{j=1}^{i} Z'_i), \qquad (2.12)$$

где $F_{Nз}$ – логическая функция, описывающая условия сохранения кибернетической безопасности ИКС с N-уровневым стеком протоколов;

X_i – логическая переменная, характеризующая факт работоспособности средств ИКС, реализующих протокол i-го уровня;

Y_i – логическая переменная (Y'_i – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на протоколе i-го уровня;

Z_i – логическая переменная (Z'_i – ее инверсия), характеризующая факт работоспособности механизма защиты информации на протоколе i-го уровня;

$F_{Nн}$ – логическая функция, описывающая условия нарушения кибернетической безопасности ИКС с N-уровневым стеком протоколов.

Для оценки важности элементов ИКС с точки зрения их влияния на сохранение кибернетической безопасности воспользуемся понятием значимости элементов системы, определенным в [102] как вероятность истинности булевой разности логической функции, описывающей условия сохранения работоспособности системы (сохранения кибернетической безопасности), по соответствующей логической переменной, характеризующей факт работоспособности соответствующего механизма защиты информации.

Согласно [102] булевой разностью функции $f(x_1, \ldots, x_n)$ по аргументу x_i называется результат логического сложения по модулю 2 исходной функции и функции, полученной из исходной путем замены аргумента x_i на его отрицание:

$$\Delta_{x_i} f(x_1,\ldots,x_n) = f(x_1,\ldots,x_i,\ldots,x_n) \oplus f(x_1,\ldots,x'_i,\ldots,x_n). \qquad (2.13)$$

Там же показано, что

$$\Delta_{x_i} f(x_1,\ldots,x_n) = f_1^{(i)}(x_1,\ldots,x_n) \oplus f_0^{(i)}(x_1,\ldots,x_n), \qquad (2.14)$$

где $f_1^{(i)}(x_1, \ldots, x_n)$ и $f_0^{(i)}(x_1, \ldots, x_n)$ – функции, полученные из исходной $f(x_1, \ldots, x_n)$ путем замены аргумента x_i на 1 и 0 соответственно.

С учетом того, что, согласно правилам алгебры логики

$$A \oplus B = AB' \vee A'B, \qquad (2.15)$$

выражение (2.14) можно записать в следующем виде:

$$\Delta_{x_i} f(x_1,\ldots,x_n) = f_1^{(i)}(x_1,\ldots,x_n) f_0^{(i)}{}'(x_1,\ldots,x_n) \vee f_1^{(i)}{}'(x_1,\ldots,x_n) f_0^{(i)}(x_1,\ldots,x_n). \qquad (2.16)$$

Учитывая (2.16), для определения значимости механизмов защиты, реализованных, соответственно, на нижнем и на верхнем уровнях стека протоколов, получим выражения булевых разностей функции $F_з$ (2.9) по переменным Z_1 и Z_2:

$$F_{з_1}^{Z_1} = X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z_2 \vee X_1 \cdot X_2 \cdot Y_1 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z'_2 =$$
$$= X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \vee X_1 \cdot X_2 \cdot Y_1 = X_1 \cdot X_2 \cdot Y'_1 \vee X_1 \cdot X_2 \cdot Y_1 =$$
$$= X_1 \cdot X_2$$

$$F_{з_0}^{Z_1} = X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z_2$$

$$F'^{Z_1}_{з_1} = X'_1 \vee X'_2$$

$$F'^{Z_1}_{з_0} = X'_1 \vee X'_2 \vee Y_1 \vee Y_2 \cdot Z'_2$$

$$\Delta_{Z_1} F_з = X_1 \cdot X_2 \cdot (X'_1 \vee X'_2 \vee Y_1 \vee Y_2 \cdot Z'_2) \vee (X'_1 \vee X'_2) \cdot (X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee$$
$$\vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z_2) = X_1 \cdot X_2 \cdot Y_1 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z'_2 \qquad (2.17)$$

$$F_{з_1}^{Z_2} = X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \vee X_1 \cdot X_2 \cdot Y_1 \cdot Z_1 =$$
$$= X_1 \cdot X_2 \cdot Y'_1 \vee X_1 \cdot X_2 \cdot Y_1 \cdot Z_1$$

$$F_{з_0}^{Z_2} = X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z_1 \vee X_1 \cdot X_2 \cdot Y_1 \cdot Z_1$$

$$F'^{Z_2}_{з_1} = X'_1 \vee X'_2 \vee Y_1 \cdot Z'_1$$

$$F'^{Z_2}_{з_0} = X'_1 \vee X'_2 \vee Y_1 \cdot Z'_1 \vee Y_2 \cdot Z'_1$$

$$\Delta_{Z_2} F_з = (X_1 \cdot X_2 \cdot Y'_1 \vee X_1 \cdot X_2 \cdot Y_1 \cdot Z_1) \cdot (X'_1 \vee X'_2 \vee Y_1 \cdot Z'_1 \vee Y_2 \cdot Z'_1) \vee$$
$$\vee (X'_1 \vee X'_2 \vee Y_1 \cdot Z'_1) \cdot (X_1 \cdot X_2 \cdot Y'_1 \cdot Y'_2 \vee X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z_1 \vee \qquad (2.18)$$
$$\vee X_1 \cdot X_2 \cdot Y_1 \cdot Z_1) = X_1 \cdot X_2 \cdot Y'_1 \cdot Y_2 \cdot Z'_1$$

Поскольку выражения (2.17) и (2.18) представляют собой ортогональные дизъюнктивные нормальные формы, допускающие, согласно [102], прямой переход к замещению, выражения для определения значимости механизмов защиты, реализованных на нижнем и на верхнем уровнях стека протоколов, полученные с учетом правил вероятностной логики, имеют вид

$$g_{Z_1} = P(\Delta_{Z_1} F_з) = P_{X_1} \cdot P_{X_2} \cdot P_{Y_1} + P_{X_1} \cdot P_{X_2} \cdot (1 - P_{Y_1}) \cdot P_{Y_2} \cdot (1 - P_{Z_2}); \qquad (2.19)$$

$$g_{Z_2} = P(\Delta_{Z_2} F_з) = P_{X_1} \cdot P_{X_2} \cdot (1 - P_{Y_1}) \cdot P_{Y_2} \cdot (1 - P_{Z_1}), \qquad (2.20)$$

где g_{Z_1} – значимость механизма защиты, реализованного на нижнем уровне стека протоколов;

g_{Z_2} – значимость механизма защиты, реализованного на верхнем уровне стека протоколов;

P_{X_1} – вероятность работоспособности средств ИКС, реализующих протокол нижнего уровня;

P_{X_2} – вероятность работоспособности средств ИКС, реализующих протокол верхнего уровня;

P_{Y_1} – вероятность реализации кибернетической угрозы на нижнем уровне стека протоколов;

P_{Y_2} – вероятность реализации кибернетической угрозы на верхнем уровне стека протоколов;

P_{Z_1} – вероятность работоспособности механизма защиты, реализованного на нижнем уровне стека протоколов;

P_{Z_2} – вероятность работоспособности механизма защиты, реализованного на протоколе верхнего уровня.

Анализ выражений (2.19) и (2.20) показывает, что при равных вероятностях работоспособности механизмов защиты, реализованных на различных уровнях стека протоколов, всегда выполняется условие

$$g_{Z_1} > g_{Z_2}. \tag{2.21}$$

С учетом того, что выражение для логической функции, описывающей условия сохранения кибернетической безопасности ИКС с двухуровневым стеком протоколов (2.9), представляет собой частный случай выражения (2.11), для системы с N уровнями стека протоколов также можно утверждать, что при равных вероятностях работоспособности механизмов защиты, реализованных на различных уровнях стека протоколов, значимость механизма защиты, реализованного на более низком уровне стека протоколов, всегда превышает значимость механизма защиты, реализованного на более высоком уровне стека протоколов.

Таким образом, результаты проведенного исследования позволяют сделать следующие выводы об особенностях реализации кибернетических угроз и функционировании механизмов защиты в ИКС с открытой архитектурой, которые должны быть учтены в разрабатываемой модели:

– процесс обработки информации в ИКС с открытой архитектурой описывается при помощи графа монотонной структуры (и соответствующей логической функции), а процессы развития ОСС при реализации кибернетических угроз и процессы предотвращения кибернетических угроз механизмами защиты – при помощи графов немонотонной структуры (и соответствующих логических функций) с множеством альтернативных путей перехода из одного состояния в другое, соответствующим различным сценариям развития ОСС и различным вариантам предотвращения кибернетических угроз;

– показатели значимости механизмов защиты, реализованных на различных уровнях стека протоколов, характеризующие важность данных механизмов с точки зрения их влияния на обеспечение кибернетической безопасности, ранжированы по убыванию от более низкого уровня стека протоколов к более высокому, т. е. в порядке, обратном нумерации уровней стеков протоколов.

2.2. Выбор параметров разрабатываемой модели защищенной ИКС

2.2.1. Выбор моделируемой характеристики и набора параметров модели

Как показано в п. 1.7, для решения задачи обеспечения кибернетической безопасности ИКС с открытой архитектурой должна быть разработана соответствующая модель функционирования СЗИ и оценки уровня обеспечения кибернетической безопасности от заданного множества кибернетических угроз. Данная модель должна строиться на основе принципов построения модели системы безопасности с полным перекрытием, с использованием аппарата ЛВТ безопасности и ее ЛВМ. В разработанной модели должны учитываться факт функционирования ИКС в условиях риска, при неизвестных законах и числовых характеристиках распределения кибернетических угроз, различная величина ущерба от реализации разных кибернетических угроз, а также особенности функционирования механизмов защиты в ИКС с открытой архитектурой. Разработанная модель должна обеспечивать возможность оценки вероятности предотвращения заданного множества кибернетических угроз выбранными механизмами защиты, а также давать возможность установить зависимость между данной вероятностью и стоимостью средств защиты, реализующих различные механизмы.

Как показано в п. 1.4, в качестве параметров модели системы безопасности с полным перекрытием рассматриваются: множество кибернетических угроз T, множество объектов ИКС O и множество механизмов защиты M, реализованных в СЗИ ИКС.

Для обеспечения возможности оценки уровня кибернетической безопасности ИКС, функционирующей в условиях риска, в качестве моделируемой характеристики необходимо использовать вероятность сохранения кибернетической безопасности, определенную, по аналогии с определением вероятности безопасного состояния ССС в классическом логико-вероятностном методе (ЛВМ)

[102, 103], как вероятность предотвращения всех включенных во множество *T* кибернетических угроз, которые могут быть реализованы по отношению ко множеству *O* объектов ИКС, реализованных в СЗИ множеством механизмов защиты *M*. При этом в качестве информационной угрозы (ИУ) нарушения безопасности выступают попытки реализации кибернетических угроз.

В соответствии с правилами ЛВМ в логическую модель (функцию опасного состояния (ФОС)) моделируемой ИКС должны быть включены в качестве параметров переменные, характеризующие факт наличия соответствующих ИУ (кибернетических угроз), реализуемых по отношению к тем или иным объектам, а также переменные, характеризующие факт работоспособности механизмов защиты с точки зрения их способности противодействовать данным угрозам информации. В процессе перехода к вероятностной модели данные переменные, по определенным правилам, должны быть заменены соответствующими вероятностными характеристиками реализации кибернетических угроз и вероятностными характеристиками предотвращения данных кибернетических угроз механизмами защиты, реализованными в СЗИ ИКС. Кроме этого, в модели должен быть обеспечен учет различной величины ущерба от реализации разных кибернетических угроз.

2.2.2. Выбор включаемых в модель характеристик объектов ИКС

Для обеспечения достоверности результатов моделирования поведения ИКС с открытой архитектурой при воздействии на нее различных кибернетических угроз необходимо в качестве элементов множества объектов *O* выбрать такие, по отношению к которым может быть достоверно оценена вероятность реализации кибернетических угроз (т. е. относительная частота попыток их реализации постоянна или изменяется в достаточно небольшом диапазоне).

С учетом приведенных в п. 1.2 особенностей функционирования ИКС с открытой архитектурой, а также на основе опубликованных результатов мониторинга и анализа инцидентов с безопасностью в сети Интернет [104, 105] можно сделать вывод о том, что в качестве элементов множества объектов *O* моделируемой ИКС, по отношению к которым рассматриваются попытки реализации множества кибернетических угроз *T*, необходимо рассматривать информационные пакеты, обрабатываемые на различных уровнях стека протоколов.

2.2.3. Выбор включаемых в модель характеристик кибернетических угроз

Как показано в п. 1.4, включаемые в разрабатываемую модель характеристики множества кибернетических угроз *T* должны учитывать вероятностный характер попыток реализации угроз, а также различную величину ущерба от реализации разных угроз.

Как показано в [106], имеется два альтернативных подхода к решению данной задачи. Первый строится на классическом принципе определения относительных частот событий при длительных испытаниях. Однако при работе с классом объектов, для которых данные частоты очень малы («редкие события»), используется альтернативный подход, основанный на идее представления вероятности как меры субъективных мнений и убеждений. Некоторые из возможных подходов к оценке вероятности проявления тех или иных угроз (эмпирическая оценка, непосредственная регистрация событий, оценка по таблице, метод «Дельфийский оракул») изложены в [106]. К сожалению, данные подходы требуют достаточно большой статистики о фактах и результатах реализации различных угроз, а также никак не учитывают цель реализации той или иной угрозы и возможный ущерб от нее.

В [107] подобный подход предлагается заменить комплексной оценкой показателя привлекательности кибернетической угрозы для нарушителя, полученной на основании оценок угрозы по следующим показателям и шкалам:

– нарушаемые принципы безопасности (1 – нарушение конфиденциальности личной, служебной и другой доверительной информации; 2 – нарушение целостности и достоверности хранимых данных с помощью специальных программ; 3 – нарушение доступности системы, данных и услуг всеми уполномоченными пользователями; 4 – несоблюдение законов, правил, лицензий, договоров и этических норм при использовании информации);

– возможность предотвращения (1 – легко; 2 – трудно; 3 – очень трудно; 4 – невозможно);

– обнаружение угрозы (1 – легко; 2 – трудно; 3 – невозможно);

– возможность нейтрализации/восстановления (1 – легко; 2 – трудно; 3 – очень трудно; 4 – невозможно);

– частота появления (0 – неизвестна; 1 – низкая; 2 – средняя; 3 – высокая; 4 – сверхвысокая);

– потенциальная опасность (1 – низкая; 2 – высокая; 3 – сверхвысокая);

– источник появления (1 – внутренний; 2 – внешний);

– уровень знаний, необходимый для подготовки и реализации соответствующей угрозы (1 – фундаментальные знания системной организации ресурсов, протоколов связи и др.; 2 – знание операционной системы; 3 – знание языков программирования; 4 – элементарные знания в области вычислительной техники);

– затраты на проектирование и разработку средств реализации угрозы (1 – большие; 2 – средние; 3 – малые);

– простота реализации (1 – очень трудно; 2 – трудно; 3 – относительно нетрудно; 4 – легко);

– потенциальное наказание в рамках существующего законодательства (1 – строгое наказание (вплоть до уголовной ответственности); 2 – незначительное наказание; 3 – наказание отсутствует).

Комплексную оценку кибернетической угрозы, отражающую ее привлекательность для злоумышленника, предлагается рассчитывать следующим образом:

$$K_i^2 = \frac{\sum_{j=1}^{m} \frac{K_{ij} + K_{j\max}}{2}}{m}, \qquad (2.21)$$

где K_i – комплексная оценка i-й угрозы;

K_{ij} – оценка i-й угрозы по j-му показателю;

m – количество параметров оценки;

$K_{j\max}$ – максимальная оценка по j-му показателю.

Данный подход представляется более обоснованным, однако он не лишен недостатков. Так, некоторые показатели (например, сложность разработки и реализации или необходимые затраты) очень трудно поддаются экспертной оценке, а кроме этого, представляется не совсем правильным оценивать комплексную оценку в виде взвешенной суммы, поскольку при этом не учитывается возможное взаимное усиление/ослабление влияния различных показателей на итоговую оценку (например, показателей, отражающих частоту появления, простоту реализации и возможный ущерб).

В связи с вышеизложенным представляется более предпочтительным в качестве характеристики множества кибернетических угроз использовать

показатель эффективности реализации кибернетических угроз, отражающий в себе как вероятностные характеристики реализации угроз, так и привлекательность угрозы с точки зрения реализации конечных целей злоумышленника (т. е. нанесения максимального ущерба владельцу обрабатываемой информации).

В теоретических и практических работах [108, 109] существует множество толкований понятия эффективности. По существу, эффективностью называют некоторое свойство или качество системы, выходящее за рамки известных групп свойств: динамических, структурных, прочностных. Несмотря на разнообразие толкований понятия эффективности, можно выделить два наиболее часто встречающихся определения:

– эффективность – это оценка использования потенциальных возможностей объекта или системы (например, применительно к автоматизированным системам управления производствами используют показатель эффективности как отношение числа правильно решенных задач к общему числу решенных задач на заданном интервале времени);

– эффективность – это оценка дополнительной отдачи управляемого объекта, полученной за счет применения системы управления, соотнесенной с полными затратами на систему. Такое понимание эффективности основывается на известном принципе экономики «затраты-результаты».

В [108] в качестве наиболее универсальной формы показателя эффективности предлагается относительная форма:

$$E(t) = \frac{G_{\Sigma}(t)}{З_{\Sigma}(t)}, \qquad (2.22)$$

где $G_{\Sigma}(t)$ – суммарная целевая отдача системы на интервале эксплуатации $(0, t)$;

$З_{\Sigma}(t)$ – полные затраты на всех этапах «жизни» системы.

В [109] для оценки эффективности различных систем предлагается подход, предполагающий подразделение систем (с точки зрения эффективности их функционирования) на два класса: системы длительного действия и системы мгновенного (точнее, кратковременного) действия.

Системы длительного действия характеризуются тем, что для выполнения операций необходимо функционирование системы в течение интервала времени $[t, t + \theta]$. Очевидно, что в этом случае качество функционирования, или выходной эффект, будет зависеть от того, как реализована траектория

$X(t,\theta)$ процесса переходов системы в этом интервале, т. е. через какие состояния в какие моменты времени проходит система.

Системы мгновенного действия характеризуются тем, что качество их функционирования полностью определяется состоянием в момент выполнения задачи, а эффективность выполнения операции в момент времени t определяется конкретным видом состояния $X(t)$.

В [109] предлагается следующий порядок расчета эффективности сложных систем мгновенного действия:

– определяется назначение действия системы;

– выясняется круг возможных задач и условий работы системы;

– оценивается ожидаемая частота повторения данных задач и условий работы;

– выбирается приемлемая в данном случае количественная мера качества функционирования системы;

– производится разбиение сложной системы на отдельные элементы;

– составляется функциональная схема системы;

– вычисляются показатели работоспособности элементов, характеризующие вероятность состояния каждого элемента в любой интересующий момент времени;

– по формуле умножения вероятностей вычисляются вероятности всех возможных состояний системы на основе вероятности состояний отдельных элементов;

– оцениваются показатели эффективности всех возможных состояний системы.

Там же показано, что для систем с аддитивными показателями эффективности отдельных элементов (т. е. таких систем, каждый из элементов которых вносит определенную и независимую долю в общий выходной эффект) суммарная эффективность может быть рассчитана по формуле

$$E(t) = \sum_i \delta_i r_i(t), \qquad (2.23)$$

где δ_i – доля, вносимая i-м элементом в суммарный выходной эффект;

$r_i(t)$ – показатель, характеризующий вероятность работоспособности i-го элемента.

Поскольку результаты реализации кибернетических угроз определяются характеристиками информации и самой угрозы, а не результатами реализации

других угроз, справедливо будет утверждать, что в качестве модели действий злоумышленника в ИКС с открытой архитектурой можно принять модель системы мгновенного действия с аддитивными показателями эффективности отдельных элементов. С учетом того, что в качестве включаемых в модель объектов ИКС выбраны информационные пакеты, обрабатываемые на различных уровнях стека протоколов, для оценки показателя эффективности реализации *i*-й кибернетической угрозы, реализуемой по отношению к информационному пакету, обрабатываемому на протоколе *j*-го уровня, предлагается использовать выражение

$$E_{ij} = Q_{ij} R_{ij}, \qquad (2.24)$$

где Q_{ij} – показатель, характеризующий долю (величину) ущерба, вносимую *i*-й кибернетической угрозой, реализуемой на протоколе *j*-го уровня, в суммарный выходной эффект действий злоумышленника;

R_{ij} – показатель, характеризующий вероятность реализации i-й кибернетической угрозы на протоколе *j*-го уровня.

Показатель Q_{ij} отражает привлекательность угрозы с точки зрения реализации конечных целей злоумышленника. Фактически показатель эффективности реализации кибернетической угрозы, представленный в виде выражения (2.24), представляет собой показатель оценки риска, связанного с реализацией конкретной кибернетической угрозы в его классическом виде [110, 111, 112, 113, 114, 115].

Таким образом, множество кибернетических угроз *T* в разрабатываемой модели должно быть представлено множеством показателей эффективности их реализации, $E = \{E_{ij}\}$, $i \in \{1, \ldots, L\}$, $j \in \{1, \ldots, N\}$, где E_{ij} – показатель эффективности реализации *i*-й кибернетической угрозы на протоколе *j*-го уровня; *L* – количество выявленных кибернетических угроз, включенных в множество угроз *T*; *N* – количество уровней стека протоколов ИКС.

2.2.4. Выбор включаемых в модель характеристик механизмов защиты

Поскольку в разрабатываемой модели в качестве элементов множества объектов *O* моделируемой ИКС, по отношению к которым рассматриваются попытки реализации множества кибернетических угроз *T*, рассматриваются информационные пакеты, обрабатываемые на различных уровнях

стека протоколов, характеристики механизмов защиты, включаемые в модель, также должны быть привязаны к уровням стека протоколов. Это значит, что в логической модели ИКС (в ФОС) наличие механизмов защиты на различных уровнях стека должно отражаться соответствующими логическими переменными.

В процессе перехода по правилам ЛВМ от ФОС к вероятностной модели необходимо будет [102] заменять логические переменные, отражающие факт наличия механизмов защиты, вероятностями их работоспособности. Вероятность работоспособности механизма защиты предлагается определить как вероятность предотвращения с его помощью соответствующей кибернетической угрозы. Данную вероятность можно определить следующим образом:

$$P_{M_{ij}} = M_{ij} K_{ij}, \qquad (2.25)$$

где M_{ij} – целочисленный параметр, значение которого (0 или 1) определяет факт наличия/отсутствия механизма защиты от i-й кибернетической угрозы на протоколе j-го уровня;

K_{ij} – коэффициент эффективности механизма защиты от i-й кибернетической угрозы на протоколе j-го уровня, характеризующий степень его надежности с точки зрения предотвращения кибернетических угроз.

Таким образом, в разрабатываемой модели множество механизмов M должно быть представлено в виде $M = \{M_{ij}\}$, $i \in \{1, \ldots, L\}$, $j \in \{1,\ldots, N\}$, где M_{ij} – целочисленный параметр, значение которого («0» или «1») определяет факт наличия/отсутствия механизма защиты от i-й угрозы на протоколе j-го уровня; L – количество выявленных кибернетических угроз, включенных в множество угроз T; N – количество уровней стека протоколов ИКС. Кроме этого, в качестве характеристик механизмов защиты в модель должно быть включено множество коэффициентов эффективности механизмов $K = \{K_{ij}\}$.

Отметим, что вопрос определения значений коэффициентов эффективности механизмов защиты K_{ij} требует отдельного рассмотрения, в большинстве случаев можно принимать $K_{ij} = 1$, т. е. считать, что в случае наличия механизма защиты он всегда обеспечивает предотвращение соответствующей кибернетической угрозы.

С учетом выбранных параметров модели задачу разработки модели защищенной ИКС с открытой архитектурой, функционирующей в условиях риска,

можно сформулировать как задачу разработки модели, обеспечивающей оценку вероятности сохранения защищенности обрабатываемой информации P как функции множества кибернетических угроз T, представленных показателями эффективности их реализации E и множества реализованных в СЗИ ИКС механизмов защиты M с коэффициентами эффективности K:

$$P = f(E, M, K). \qquad (2.26)$$

Выводы

Выполнено исследование особенностей ИКС с открытой архитектурой с точки зрения их кибернетической безопасности, выявлены следующие закономерности их функционирования: множество альтернативных путей развития опасного состояния при реализации кибернетических угроз и ранжирование показателей значимости механизмов защиты, реализованных на различных уровнях стека протоколов.

Определены, с учетом условий и закономерностей функционирования ИКС, структура и набор параметров математической модели, защищенной ИКС. Обоснована возможность использования в качестве моделируемой характеристики вероятности сохранения кибернетической безопасности. Для учета выявленных закономерностей функционирования систем с открытой архитектурой обоснована возможность рассмотрения в качестве элементов множества объектов моделируемой системы информационных пакетов, обрабатываемых на различных уровнях стека протоколов.

В отличие от классического ЛВМ, обоснована возможность представления в разрабатываемой модели множества кибернетических угроз не вероятностями, а показателями эффективности, характеризующими степень риска, связанного с реализацией угроз на различных уровнях стека протоколов.

3. ИЕРАРХИЧЕСКАЯ ЛОГИКО-ВЕРОЯТНОСТНАЯ МОДЕЛЬ ЗАЩИЩЕННОЙ ИКС С ОТКРЫТОЙ АРХИТЕКТУРОЙ

В данной главе, используя ЛВМ, разработаем математическую модель защищенной ИКС с открытой архитектурой. Для этого построим модели ИКС с двухуровневым и трехуровневым стеками протоколов при реализации одной кибернетической угрозы, проанализируем их особенности и обобщим на случай системы с произвольным стеком протоколов для произвольного множества угроз.

Выполним вычислительное исследование разработанной математической модели, определим, насколько правильно в ней отражены выявленные условия и закономерности функционирования ИКС с открытой архитектурой.

3.1. Алгоритм построения логико-вероятностной модели защищенной ИКС с открытой архитектурой

Для решения задачи разработки модели защищенной ИКС необходимо в соответствии с требованиями ЛВМ разработать логические и вероятностные модели поведения ИКС с открытой архитектурой, учитывающие показанные в п. 2.1 особенности их функционирования, а также определенный в п. 2.2 набор параметров, т. е. необходимо, согласно методике, изложенной в [116], с учетом указанных особенностей: выполнить содержательное описание модели поведения ИКС; описать сценарий развития ОСС на языке продукций (логических высказываний); построить граф-модель развития ОСС исследуемой системы, позволяющую осуществить событийно-логическое описание ее функционирования; выполнить определение логической ФОС; преобразовать ФОС в форму, позволяющую выполнять непосредственные расчеты вероятностных характеристик кибернетической безопасности ИКС.

3.2. Построение модели защищенной ИКС с двухуровневым стеком протоколов

3.2.1. Содержательное описание поведения ИКС с двухуровневым стеком протоколов

Содержательное описание поведения ИКС с двухуровневым стеком протоколов при реализации одной кибернетической угрозы имеет вид, представленный ниже.

Успешная реализация кибернетической угрозы в ИКС с двухуровневым стеком протоколов произойдет в случае, если:

1) отсутствует попытка реализации угрозы на верхнем уровне стека протоколов, имеется попытка реализации угрозы на нижнем уровне стека протоколов и отсутствуют механизмы защиты на нижнем уровне стека протоколов;

2) имеется попытка реализации угрозы на верхнем уровне стека протоколов, отсутствует попытка реализации угрозы на нижнем уровне стека протоколов и отсутствуют механизмы защиты как на верхнем, так и на нижнем уровне стека протоколов;

3) имеется попытка реализации угрозы как на верхнем, так и на нижнем уровне стека протоколов и, независимо от наличия или отсутствия механизмов защиты на верхнем уровне стека протоколов, отсутствуют механизмы защиты на нижнем уровне стека протоколов.

3.2.2. Описание сценария развития опасного состояния в ИКС с двухуровневым стеком протоколов

Для описания сценария развития ОСС на языке продукций (логических высказываний) выделим в выше приведенном содержательном описании модели поведения:

1) факты-события, связанные с наличием/отсутствием попыток реализации кибернетических угроз заглавными буквами X/X' (и идентификаторами с индексами для использования в граф-схеме);

2) факты-события, связанные с наличием/отсутствием механизмов защиты заглавными буквами Y/Y' (и идентификаторами с индексами для использования в граф-схеме);

3) факты-события, связанные с успешной реализацией кибернетических угроз заглавными буквами F (и идентификаторами с индексами для использования в граф-схеме);

4) союзы «и», «или» как отношения заглавными буквами «И», «ИЛИ».

Сценарий развития ОСС на языке продукций для ИКС с двухуровневым стеком протоколов имеет следующий вид:

Правило 1. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ (F) имеет место:

– ЕСЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (F_1);

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (F_2);

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ОБОИХ УРОВНЯХ (F_{12}).

Правило 2. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (F_1) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1);

– И нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_2');

– И отсутствуют МЕХАНИЗМЫ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1').

Правило 3. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (F_2) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1');

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_2);

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА ВЕРХНЕМ УРОВНЕ (Y_2').

Правило 4. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ОБОИХ УРОВНЯХ (F_{12}) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1);

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_2);

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1').

3.2.3. Построение граф-модели развития опасного состояния и определение логической функции опасности системы

Структурная граф-схема (граф-модель) развития ОСС для ИКС с двухуровневым стеком протоколов, построенная с использованием аппарата СФЦ, приведена на рис. 3.1.

На основании построенной граф-модели получим выражение для логической ФОС ИКС с двухуровневым стеком протоколов:

$$F = F_1 \vee F_2 \vee F_{12} = X_1 X_2 ' Y_1 ' \vee X_1 ' X_2 Y_1 ' Y_2 ' \vee X_1 X_2 Y_1 ' = X_1 Y_1 ' (X_2 ' \vee X_2) \vee X_1 ' X_2 Y_1 ' Y_2 ' = X_1 Y_1 ' \vee X_1 ' X_2 Y_1 ' Y_2 '. \quad (3.1)$$

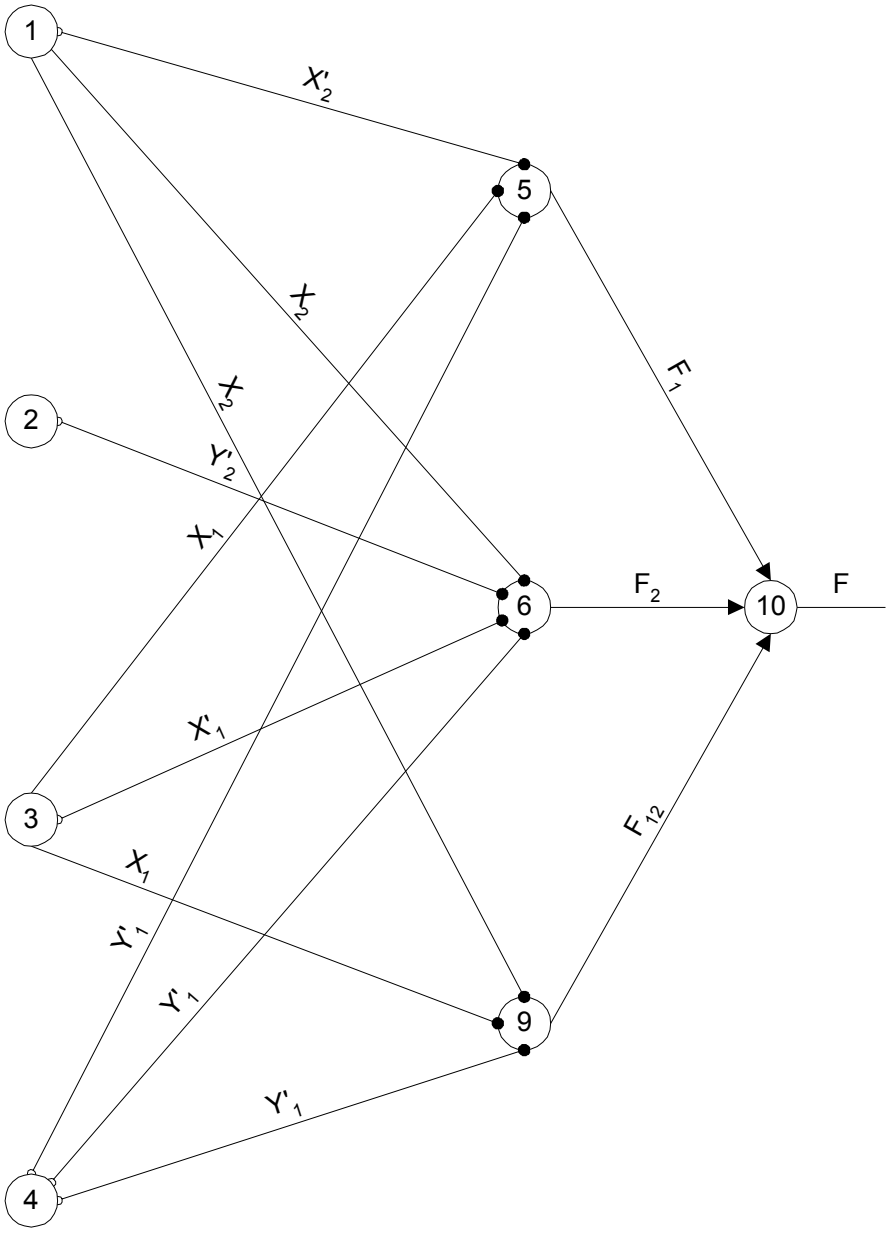


Рис. 3.1. Граф-модель развития ОСС для ИКС с двухуровневым стеком протоколов

Таким образом, окончательное выражение для логической ФОС ИКС с двухуровневым стеком протоколов

$$F = X_1 Y_1' \vee X_1' X_2 Y_1' Y_2', \tag{3.2}$$

где X_1 – переменная (X_1' – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на нижнем уровне стека протоколов (0 – нет угрозы, 1 – есть угроза);

X_2 – переменная (X_2' – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на верхнем уровне стека протоколов (0 – нет угрозы, 1 – есть угроза);

Y_1 – переменная (Y_1' – ее инверсия), характеризующая факт наличия механизма защиты на нижнем уровне стека протоколов (0 – нет, 1 – есть);

Y_2 – переменная (Y_2' – ее инверсия), характеризующая факт наличия механизма защиты на верхнем уровне стека протоколов (0 – нет, 1 – есть).

3.2.4. Определение вероятностной функции оценки уровня кибернетической безопасности ИКС с двухуровневым стеком протоколов

Полученное выражение в виде (3.2) представляет собой дизъюнктивную нормальную форму с взаимно ортогональными членами и, согласно доказанной в вероятностной логике теореме замещения [102], может быть преобразовано в функцию вычисления вероятности реализации опасного события путем замены логических переменных вероятностями их реализации, отрицаний логических переменных – на 1 минус вероятность реализации, логических сложений и логических умножений их арифметическими аналогами. Учтем, что в нашем случае вместо вероятности реализации кибернетической угрозы используется значение показателя эффективности реализации угрозы, а также то, что вероятность предотвращения кибернетической угрозы механизмом защиты определяется наличием такого механизма и его коэффициентом эффективности. Обозначив через E_i показатель эффективности реализации кибернетической угрозы на i-м уровне протокола, через M_i – параметр, значение которого («0» или «1») определяет наличие механизма защиты, реализованного на i-м уровне протокола, а через K_i – коэффициент эффективности механизма M_i, получим следующее выражение вероятности реализации ОСС:

$$T_2 = E_1(1 - M_1 K_1) + (1 - E_1) E_2 (1 - M_1 K_1)(1 - M_2 K_2). \tag{3.3}$$

Получим выражение для вероятностной функции оценки уровня кибернетической безопасности ИКС с двухуровневым стеком протоколов:

$$P_2 = 1 - E_1(1 - M_1K_1) - (1 - E_1)E_2(1 - M_1K_1)(1 - M_2K_2) =$$
$$= (1 - E_1)(1 - E_2) + E_1M_1K_1 + (1 - E_1)E_2(M_1K_1(1 - M_2K_2) + M_2K_2).$$

Таким образом, окончательное выражение для вероятностной функции оценки кибернетической безопасности ИКС с двухуровневым стеком протоколов имеет вид

$$P_2 = (1 - E_1)(1 - E_2) + E_1M_1K_1 + (1 - E_1)E_2(M_1K_1(1 - M_2K_2) + M_2K_2), \quad (3.4)$$

где E_1 – показатель эффективности реализации кибернетической угрозы на нижнем уровне стека протоколов;

E_2 – показатель эффективности реализации кибернетической угрозы на верхнем уровне стека протоколов;

M_1 – параметр, значение которого («0» или «1») определяет наличие механизма защиты, реализованного на нижнем уровне стека протоколов;

K_1 – коэффициент эффективности механизма защиты, реализованного на нижнем уровне стека протоколов;

M_2 – параметр, значение которого («0» или «1») определяет наличие механизма защиты, реализованного на верхнем уровне стека протоколов;

K_2 – коэффициент эффективности механизма защиты, реализованного на верхнем уровне стека протоколов.

3.3. Построение модели защищенной ИКС с трехуровневым стеком протоколов

3.3.1. Содержательное описание поведения ИКС с трехуровневым стеком протоколов

Содержательное описание ИКС с трехуровневым стеком протоколов при реализации одной кибернетической угрозы имеет вид, представленный ниже.

Успешная реализация кибернетической угрозы в ИКС с трехуровневым стеком протоколов произойдет в случае, если:

1) отсутствует попытка реализации угрозы на верхнем и на среднем уровнях стека протоколов, имеется попытка реализации угрозы на нижнем уровне стека протоколов и отсутствует механизм защиты на нижнем уровне стека протоколов;

2) отсутствует попытка реализации угрозы на верхнем и на нижнем уровнях стека протоколов, имеется попытка реализации угрозы на среднем уровне стека протоколов и отсутствуют механизмы защиты на среднем и нижнем уровнях стека протоколов;

3) отсутствует попытка реализации угрозы на среднем и на нижнем уровнях стека протоколов, имеется попытка реализации угрозы на верхнем уровне стека протоколов и отсутствуют механизмы защиты на верхнем, среднем и нижнем уровнях стека протоколов;

4) отсутствует попытка реализации угрозы на верхнем уровне стека протоколов, имеется попытка реализации угрозы на среднем и на нижнем уровнях стека протоколов и, независимо от наличия или отсутствия механизма защиты на среднем уровне стека протоколов, отсутствует механизм защиты на нижнем уровне стека протоколов;

5) отсутствует попытка реализации угрозы на среднем уровне стека протоколов, имеется попытка реализации угрозы на верхнем и на нижнем уровнях стека протоколов и, независимо от наличия или отсутствия механизмов защиты на верхнем и среднем уровнях стека протоколов, отсутствует механизм защиты на нижнем уровне стека протоколов;

6) имеется попытка реализации угрозы на верхнем и среднем уровнях стека протоколов, отсутствует попытка реализации угрозы на нижнем уровне стека протоколов и, независимо от наличия или отсутствия механизма защиты на верхнем уровне стека протоколов, отсутствуют механизмы защиты на среднем и нижнем уровнях стека протоколов;

7) имеется попытка реализации угрозы на верхнем, среднем и нижнем уровнях стека протоколов и, независимо от наличия или отсутствия механизмов защиты на верхнем и среднем уровнях стека протоколов, отсутствует механизм защиты на нижнем уровне стека протоколов.

3.3.2. Описание сценария развития опасного состояния в ИКС с трехуровневым стеком протоколов

Сценарий развития ОСС на языке продукций для ИКС с трехуровневым стеком протоколов имеет следующий вид:

Правило 1. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ (F) имеет место:

– ЕСЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (F_1);

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (F_2);

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (F_3);

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ И СРЕДНЕМ УРОВНЯХ (F_{12});

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ И ВЕРХНЕМ УРОВНЯХ (F_{13});

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА СРЕДНЕМ И ВЕРХНЕМ УРОВНЯХ (F_{23});

– ИЛИ имеет место УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ВСЕХ УРОВНЯХ (F_{123}).

Правило 2. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (F_1) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1);

– И нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (X_2');

– И нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_3');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1').

Правило 3. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (F_2) имеет место:

– ЕСЛИ нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1');

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (X_2);

– И нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_3');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА СРЕДНЕМ УРОВНЕ (Y_2').

Правило 4. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (F_3) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1');

– И нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (X_2');

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_3);

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА СРЕДНЕМ УРОВНЕ (Y_2');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА ВЕРХНЕМ УРОВНЕ (Y_3').

Правило 5. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ И СРЕДНЕМ УРОВНЯХ (F_{12}) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1);

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (X_2);

– И нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_3');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1').

Правило 6. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА НИЖНЕМ И ВЕРХНЕМ УРОВНЯХ (F_{13}) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1);

– И нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (X_2');

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_3);

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1').

Правило 7. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА СРЕДНЕМ И ВЕРХНЕМ УРОВНЯХ (F_{23}) имеет место:

– ЕСЛИ нет ПОПЫТКИ РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1');

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (X_2);

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_3);

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1');

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА СРЕДНЕМ УРОВНЕ (Y_2').

Правило 8. УСПЕШНАЯ РЕАЛИЗАЦИЯ УГРОЗЫ НА ВСЕХ УРОВНЯХ (F_{123}) имеет место:

– ЕСЛИ есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА НИЖНЕМ УРОВНЕ (X_1);

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА СРЕДНЕМ УРОВНЕ (X_2);

– И есть ПОПЫТКА РЕАЛИЗАЦИИ УГРОЗЫ НА ВЕРХНЕМ УРОВНЕ (X_3);

– И отсутствует МЕХАНИЗМ ЗАЩИТЫ НА НИЖНЕМ УРОВНЕ (Y_1').

3.3.3. Построение граф-модели развития опасного состояния и определение логической функции опасности системы

Структурная граф-схема (граф-модель) развития ОСС для ИКС с трехуровневым стеком протоколов, построенная с использованием аппарата СФЦ, приведена на рис. 3.2.

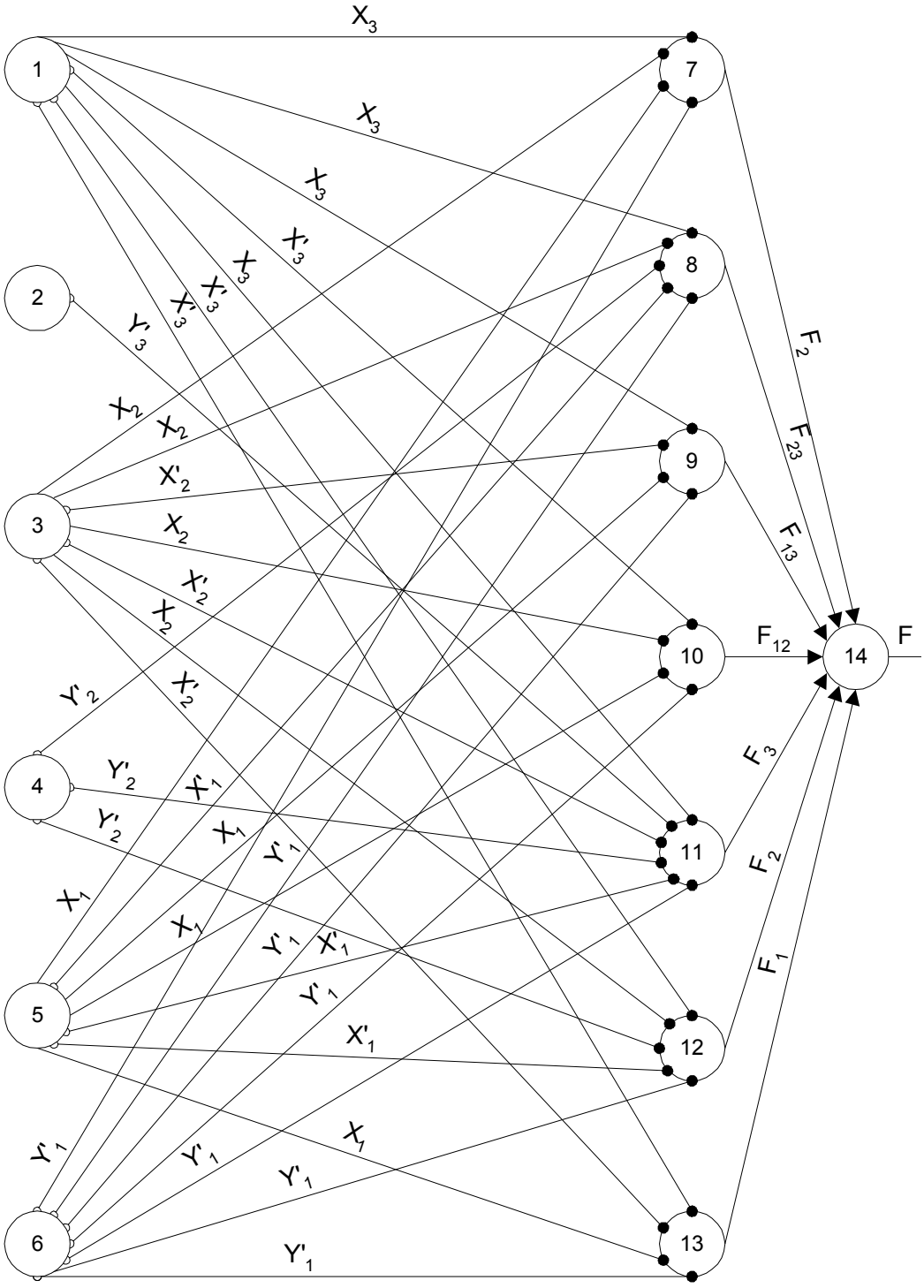


Рис. 3.2. Граф-модель развития ОСС для ИКС с трехуровневым стеком протоколов

Получим выражение для логической ФОС ИКС с трехуровневым стеком протоколов:

$$F = F_1 \vee F_2 \vee F_3 \vee F_{12} \vee F_{13} \vee F_{23} \vee F_{123} = X_1X_2'X_3'Y_1' \vee X_1'X_2X_3'Y_1'Y_2' \vee$$
$$\vee X_1'X_2'X_3Y_1'Y_2'Y_3' \vee X_1X_2X_3'Y_1' \vee X_1X_2'X_3Y_1' \vee X_1'X_2X_3Y_1'Y_2' \vee X_1X_2X_3Y_1' =$$
$$= X_1X_2'Y_1' \vee X_1'X_2Y_1'Y_2' \vee X_1'X_2'X_3Y_1'Y_2'Y_3' \vee X_1X_2Y_1' =$$
$$= X_1Y_1' \vee X_1'X_2Y_1'Y_2' \vee X_1'X_2'X_3Y_1'Y_2'Y_3'.$$

Таким образом, окончательное выражение для логической ФОС ИКС с трехуровневым стеком протоколов

$$F = X_1Y_1' \vee X_1'X_2Y_1'Y_2' \vee X_1'X_2'X_3Y_1'Y_2'Y_3', \qquad (3.5)$$

где X_1 – переменная (X_1' – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на нижнем уровне стека протоколов (0 – нет угрозы, 1 – есть угроза);

X_2 – переменная (X_2' – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на среднем уровне стека протоколов (0 – нет угрозы, 1 – есть угроза);

X_3 – переменная (X_2' – ее инверсия), характеризующая факт наличия попытки реализации кибернетической угрозы на верхнем уровне стека протоколов (0 – нет угрозы, 1 – есть угроза);

Y_1 – переменная (Y_1' – ее инверсия), характеризующая факт наличия механизма защиты на нижнем уровне стека протоколов (0 – нет, 1 – есть);

Y_2 – переменная (Y_2' – ее инверсия), характеризующая факт наличия механизма защиты на среднем уровне стека протоколов (0 – нет, 1 – есть);

Y_3 – переменная (Y_2' – ее инверсия), характеризующая факт наличия механизма защиты на верхнем уровне стека протоколов (0 – нет, 1 – есть).

3.3.4. Определение вероятностной функции оценки уровня кибернетической безопасности ИКС с трехуровневым стеком протоколов

Полученное выражение в виде (3.5), как и выражение (3.2), представляет собой дизъюнктивную нормальную форму со взаимно ортогональными членами и может, по указанным выше правилам, быть преобразована в функцию вычисления вероятности реализации опасного события. Обозначив через E_i показатель эффективности реализации угрозы информации на i-м уровне протокола, через M_i – параметр, значение которого («0» или «1») определяет

наличие механизма защиты, реализованного на *i*-м уровне протокола, а через K_i – коэффициент эффективности механизма M_i, получим следующее выражение вероятности реализации ОСС в ИКС с трехуровневым стеком протоколов:

$$T_3 = E_1(1 - M_1K_1) + (1 - E_1)E_2(1 - M_1K_1)(1 - M_2K_2) + \\ +(1 - E_1)(1 - E_2)E_3(1 - M_1K_1)(1 - M_2K_2)(1 - M_3K_3). \quad (3.6)$$

Получим выражение для вероятностной функции оценки уровня кибернетической безопасности ИКС с трехуровневым стеком протоколов:

$$P_3 = 1 - T_3 = 1 - E_1(1 - M_1K_1) - (1 - E_1)E_2(1 - M_1K_1)(1 - M_2K_2) - \\ -(1 - E_1)(1 - E_2)E_3(1 - M_1K_1)(1 - M_2K_2)(1 - M_3K_3) = \\ = (1 - E_1)(1 - E_2)(1 - E_3) + E_1M_1K_1 + (1 - E_1)E_2(M_1K_1(1 - M_2K_2) + M_2K_2) + \\ +(1 - E_1)(1 - E_2)E_3(M_1K_1(1 - M_2K_2)(1 - M_3K_3) + M_2K_2(1 - M_3K_3) + M_3K_3).$$

Таким образом, окончательное выражение для вероятностной функции оценки уровня кибернетической безопасности ИКС с трехуровневым стеком протоколов имеет вид

$$P_3 = (1 - E_1)(1 - E_2)(1 - E_3) + E_1M_1K_1 + (1 - E_1)E_2(M_1K_1(1 - M_2K_2) + M_2K_2) + \\ +(1 - E_1)(1 - E_2)E_3(M_1K_1(1 - M_2K_2)(1 - M_3K_3) + M_2K_2(1 - M_3K_3) + M_3K_3), \quad (3.7)$$

где E_1 – показатель эффективности реализации кибернетической угрозы на нижнем уровне стека протоколов;

E_2 – показатель эффективности реализации кибернетической угрозы на среднем уровне стека протоколов;

E_3 – показатель эффективности реализации кибернетической угрозы на верхнем уровне стека протоколов;

M_1 – параметр, значение которого (0 или 1) определяет наличие механизма защиты, реализованного на нижнем уровне стека протоколов;

K_1 – коэффициент эффективности механизма защиты, реализованного на нижнем уровне стека протоколов;

M_2 – параметр, значение которого (0 или 1) определяет наличие механизма защиты, реализованного на среднем уровне стека протоколов;

K_2 – коэффициент эффективности механизма защиты, реализованного на среднем уровне стека протоколов;

M_3 – параметр, значение которого (0 или 1) определяет наличие механизма защиты, реализованного на верхнем уровне стека протоколов;

K_3 – коэффициент эффективности механизма защиты, реализованного на верхнем уровне стека протоколов.

3.4. Определение вероятностной функции оценки уровня кибернетической безопасности ИКС с открытой архитектурой

Как видно из анализа выражений (3.3) и (3.6), выражение для вычисления вероятности реализации ОСС имеет явно выраженный рекуррентный характер и легко может быть обобщено на случай системы с N уровнями стека протоколов:

$$T_i(M) = \sum_{j=1}^{N}\left(E_j \prod_{k=1}^{j-1}(1-E_k)\prod_{k=1}^{j}(1-M_k K_k) \right), \qquad (3.8)$$

где N – количество уровней стека протоколов ИКС;

E_k – показатель эффективности реализации кибернетической угрозы на протоколе k-го уровня;

M_k – параметр, значение которого (0 или 1) определяет наличие механизма защиты, реализованного на протоколе k-го уровня;

K_k – коэффициент эффективности механизма защиты, реализованного на протоколе k-го уровня.

Аналогичным образом можно обобщить на случай ИКС с N уровнями стека протоколов выражение для функции оценки уровня кибернетической безопасности:

$$P_i(M) = \prod_{j=1}^{N}(1-E_j) + \sum_{j=1}^{N}\left[E_j \prod_{k=1}^{j-1}(1-E_k)\left(\sum_{k=1}^{j} M_k K_k \prod_{l=k+1}^{j}(1-M_l K_l) \right) \right]. \qquad (3.9)$$

Для ИКС с N уровнями стека протоколов и множества T кибернетических угроз, включающего L угроз, согласно правилу вычисления вероятности сложного составного события, состоящего из набора независимых событий [117], выражение для функции оценки уровня кибернетической безопасности ИКС с открытой архитектурой будет иметь следующий вид:

$$P(M) = \prod_{i=1}^{L}\left(\prod_{j=1}^{N}(1-E_{ij}) + \sum_{j=1}^{N}\left[E_{ij} \prod_{k=1}^{j-1}(1-E_{ik})\left(\sum_{k=1}^{j} M_{ik} K_{ik} \prod_{l=k+1}^{j}(1-M_{il} K_{il}) \right) \right] \right), \qquad (3.10)$$

где L – количество кибернетических угроз, включенных во множество угроз T;

N – количество уровней стека протоколов ИКС;

E_{ij} – показатель эффективности реализации i-й кибернетической угрозы на протоколе j-го уровня;

M_{ij} – параметр, значение которого («0» или «1») определяет наличие механизма защиты от i-й кибернетической угрозы, реализованного на протоколе j-го уровня;

K_{ij} – коэффициент эффективности механизма защиты от i-й кибернетической угрозы, реализованного на протоколе j-го уровня.

3.5. Исследование разработанной модели защищенной ИКС

Для проверки конструктивности разработанной модели проанализируем, насколько в полученной вероятностной модели отражены показанные в п. 2.2 особенности функционирования механизмов защиты в ИКС с открытой архитектурой, т. е. ранжирование по убыванию от более низкого уровня стека протоколов к более высоким показателям значимости механизмов защиты. Для этого исследуем влияние на уровень кибернетической безопасности вероятностей работоспособности механизмов защиты информации M_{ij}, реализованных на различных уровнях стека протоколов ИКС. При этом учтем, что, поскольку в выражение для определения уровня кибернетической безопасности ИКС (3.10) в качестве сомножителей входят одинаковые по форме функции оценки уровня кибернетической безопасности при воздействии одной i-й угрозы (3.9), характер функции (3.10) полностью определяется характером функций-сомножителей (3.9).

Для исследования зависимости функций-сомножителей выражения (3.9) от вероятности работоспособности механизма M_j воспользуемся тем, что, как показано в [102], выражение для определения значимости механизма защиты как вероятности истинности булевой разности логической функции, описывающей условия сохранения кибернетической безопасности, по соответствующей логической переменной, характеризующей факт работоспособности соответствующего механизма защиты информации, эквивалентно следующему выражению:

$$g_{M_j} = \frac{\partial P_i(M)}{\partial P_{M_j}}, \tag{3.11}$$

где, согласно (2.25), $P_{M_j} = M_j K_j$.

С учетом (3.9) получаем:

$$g_{M_j} = \sum_{k=j}^{N} E_k \prod_{l=1}^{k-1} (1 - E_l) \prod_{l=1}^{j-1} (1 - M_l K_l) \prod_{l=j+1}^{k} (1 - M_l K_l). \qquad (3.12)$$

В табл. 3.1 приведены значения полученных согласно (3.12) значимостей механизмов защиты, реализованных на j-м уровне стека протоколов, для ИКС с семиуровневым стеком протоколов при равных значениях показателей эффективности реализации кибернетических угроз и значениях $M_l = 0$, $K_l = 1$ (вариант 1).

Таблица 3.1

Исходные данные и результаты варианта 1

Уровень стека протоколов *j*	1	2	3	4	5	6	7
Значение E_j	0,14285	0,14285	0,14285	0,14285	0,14285	0,14285	0,14285
Значение g_{Mj}	0,66008	0,51722	0,39477	0,28982	0,19985	0,12274	0,05665

В табл. 3.2 приведены значения полученных согласно (3.12) значимостей механизмов защиты, реализованных на j-м уровне стека протоколов, для ИКС с семиуровневым стеком протоколов при квадратично возрастающих в зависимости от уровня стека протокола значениях показателей эффективности реализации кибернетических угроз и значениях $M_l = 0$, $K_l = 1$ (вариант 2).

Таблица 3.2

Исходные данные и результаты варианта 2

Уровень стека протоколов *j*	1	2	3	4	5	6	7
Значение E_j	0,00714	0,02857	0,06428	0,11428	0,17857	0,25714	0,35000
Значение g_{Mj}	0,68295	0,67581	0,64744	0,58544	0,48229	0,33955	0,17071

На рис. 3.3 приведена диаграмма, построенная на основании полученных по исходным данным вариантов 1 и 2 значений.

Анализ приведенных в табл. 3.1 и 3.2 результатов показывает, что, независимо от закона изменения показателей эффективности реализации кибернетических угроз, наибольшую значимость имеют механизмы защиты информации, реализованные на более низких уровнях стека протоколов. Этот результат согласуется с показанным в п. 2.1 ранжированием показателей значимости механизмов защиты, реализованных на различных уровнях стека протоколов, с точки зрения их влияния на уровень кибернетической безопасности и позволяет сделать

вывод о конструктивности разработанной модели и возможности ее применения для решения задачи построения системы обеспечения кибернетической безопасности ИКС с открытой архитектурой.

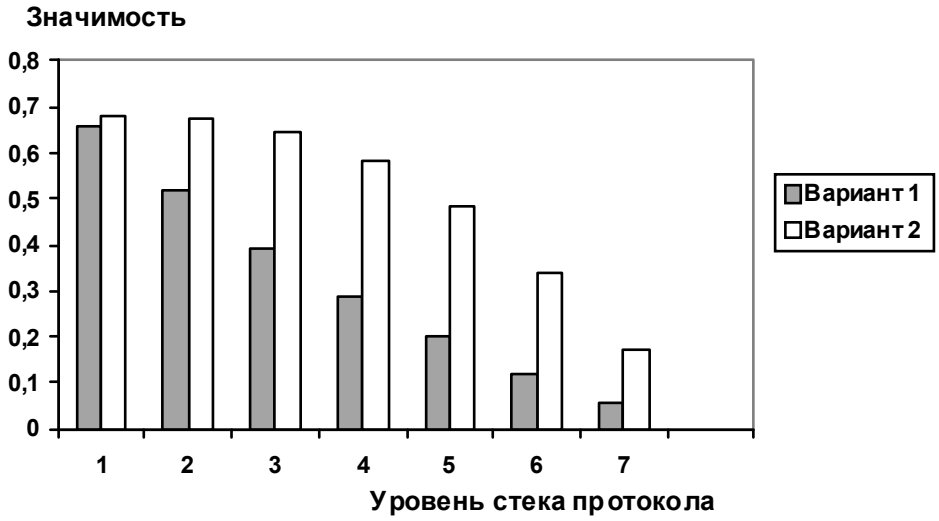


Рис. 3.3. Значения показателей значимости в ИКС с семиуровневым стеком протоколов

Выводы

С использованием ЛВМ разработана математическая модель защищенной ИКС, позволяющая оценивать уровень кибернетической безопасности ИКС при воздействии на нее заданного множества кибернетических угроз, отличающаяся учетом условий и выявленных закономерностей функционирования ИКС с открытой архитектурой. В процессе разработки модели последовательно выполнено содержательное описание поведения системы, описан сценарий развития опасного состояния на языке продукций, построены граф-модели развития опасного состояния, определены логические функции опасности, осуществлен переход к вероятностным моделям ИКС с двухуровневым и трехуровневым стеками протоколов при воздействии на них одной кибернетической угрозы. Полученные вероятностные модели обобщены на случай системы с произвольным стеком протоколов для произвольного множества кибернетических угроз.

Выполнены вычислительные исследования, результаты которых подтвердили конструктивность разработанной математической модели, правильное отражение в ней выявленных условий и закономерностей функционирования ИКС с открытой архитектурой при реализации кибернетических угроз и их предотвращении механизмами защиты.

4. ОПТИМАЛЬНОЕ РАЗМЕЩЕНИЕ МЕХАНИЗМОВ ЗАЩИТЫ ПО УРОВНЯМ СТЕКА ПРОТОКОЛОВ ВЗАИМОДЕЙСТВИЯ КОМПОНЕНТОВ ИКС

В данной главе сформулируем задачу построения оптимальной системы обеспечения кибернетической безопасности с учетом ограничений на стоимость средств защиты как задачу нелинейного целочисленного программирования с одним ограничением. Обоснуем выбор метода ее решения. Разработаем алгоритм решения сформулированной задачи, в котором учитываются выявленные закономерности функционирования механизмов защиты в ИКС с открытой архитектурой.

Выполним вычислительное исследование разработанного алгоритма, оценим его сходимость и вычислительную эффективность.

4.1. Формулировка решаемой задачи математического программирования

При решении сформулированных в п. 1.7 задач синтеза структуры и параметров оптимальной СЗИ для обеспечения кибернетической безопасности с фиксированной стоимостью средств защиты в качестве целевой функции (ЦФ), подлежащей максимизации путем решения соответствующей задачи математического программирования, предлагается использовать полученное с использованием разработанной модели защищенной ИКС выражение для оценки вероятности сохранения кибернетической безопасности ИКС с открытой архитектурой:

$$P(M)=\prod_{i=1}^{L}\left(\prod_{j=1}^{N}(1-E_{ij})+\sum_{j=1}^{N}\left[E_{ij}\prod_{k=1}^{j-1}(1-E_{ik})\left(\sum_{k=1}^{j}M_{ik}K_{ik}\prod_{l=k+1}^{j}(1-M_{il}K_{il})\right)\right]\right), \quad (4.1)$$

где L – количество кибернетических угроз, включенных в множество угроз T;

N – количество уровней стека протоколов ИКС;

E_{ij} – показатель эффективности реализации i-й кибернетической угрозы на протоколе j-го уровня;

M_{ij} – параметр, значение которого (0 или 1) определяет наличие механизма защиты от i-й кибернетической угрозы, реализованного на протоколе j-го уровня;

K_{ij} – коэффициент эффективности механизма защиты от *i*-й кибернетической угрозы, реализованного на протоколе *j*-го уровня.

Поскольку целью оптимизации является поиск решения, приемлемого по суммарной стоимости средств защиты, максимизация ЦФ должна выполняться с учетом показанного ограничения на стоимость реализованных средств защиты:

$$C(M) = \sum_{i=1}^{L}\left(\sum_{j=1}^{N} M_{ij} \cdot C_{ij}\right) \le C_o, \tag{4.2}$$

где $C(M)$ – стоимость реализации множества механизмов защиты, $M = \{M_{ij}\}$, $i \in \{1,\ldots, L\}, j \in \{1, \ldots, N\}$;

L – количество кибернетических угроз, включенных в множество угроз T;

N – количество уровней стека протоколов ИКС;

M_{ij} – параметр, значение которого («0» или «1») определяет факт наличия/отсутствия механизма защиты от *i*-й угрозы на протоколе *j*-го уровня;

C_{ij} – стоимость реализации механизма защиты от *i*-й кибернетической угрозы на протоколе *j*-го уровня;

C_o – ограничение на суммарную стоимость средств защиты.

С учетом вышесказанного постановка задачи нелинейного целочисленного программирования, которую необходимо решить для построения оптимальной системы обеспечения кибернетической безопасности ИКС с открытой архитектурой, имеет вид

$$\begin{cases} P(M) \to \max \\ C(M) \le C_o \\ M = \{M_{ij}\}, \;\; M_{ij} \in \{0,1\}. \end{cases}$$

4.2. Исследование целевой функции и выбор метода решения

Так как представленная в виде (4.1) ЦФ не является линейной, задача поиска максимума данной функции представляет собой задачу нелинейного программирования (НЛП).

При выборе метода решения задачи НЛП [118] исходят из вида ЦФ, дифференцируемости ЦФ (возможности аналитического определения частных производных ЦФ первого и более высоких порядков) и вида ограничений. Для случаев, когда вычисление частных производных ЦФ высоких порядков

невозможно или затруднено, наиболее широко применяются градиентные методы. Однако успешное применение градиентных методов [119, 120] возможно только в случае отсутствия у ЦФ локальных экстремумов, расположенных в области допустимых решений. Для выяснения возможности использования при решении поставленной задачи градиентного метода исследуем поведение ЦФ (4.1) в области допустимых решений.

Функция (4.1) является дифференцируемой по компонентам вектора изменяемых параметров (переменным M_{ij}). Выражение для определения соответствующих частных производных имеет вид

$$\frac{\partial P(M)}{\partial M_{ij}} = \frac{\partial P_i(M)}{\partial M_{ij}} \prod_{k=1}^{i-1} P_k(M) \prod_{k=i+1}^{N} P_k(M), \tag{4.3}$$

где

$$\frac{\partial P_i(M)}{\partial M_{ij}} = K_{ij} \sum_{k=j}^{N} E_{ik} \prod_{l=1}^{k-1} (1 - E_{il}) \prod_{l=1}^{j-1} (1 - M_{il} K_{il}) \prod_{l=j+1}^{k} (1 - M_{il} K_{il}); \tag{4.4}$$

$$P_k(M) = \prod_{l=1}^{N} (1 - E_{kl}) + \sum_{l=1}^{N} \left[E_{kl} \prod_{m=1}^{l-1} (1 - E_{km}) \left(\sum_{m=1}^{l} M_{km} K_{km} \prod_{n=m+1}^{l} (1 - M_{kn} K_{kn}) \right) \right]. \tag{4.5}$$

Известно [118], что наличие локальных экстремумов ЦФ определяется изменением знака частных производных. Проанализируем характер и знак частных производных ЦФ (4.1), заданных в виде (4.3), в области $M_{ij} \in [0,1]$, учитывая, что $E_{ij} \in [0,1]$ и $K_{ij} \in [0,1]$. Поскольку (4.5) представляет собой выражение для оценки вероятности сохранения кибернетической безопасности при воздействии одной k-й угрозы, а, по определению, вероятности $0 \le P_k(M) \le 1$, причем, как видно из выражения (4.5), $P_k(M)$ не зависит от M_{ij}, справедливо будет утверждать, что знак частной производной $\frac{\partial P(M)}{\partial M_{ij}}$ определяется знаком частной производной $\frac{\partial P_i(M)}{\partial M_{ij}}$. Учитывая, что в (4.4) $M_{il} \in \{0,1\}$, а $0 \le E_{ik} \le 1$, $0 \le E_{il} \le 1$, $0 \le K_{ik} \le 1$ и $0 \le K_{il} \le 1$, можно утверждать, что $\frac{\partial P_i(M)}{\partial M_{ij}} \ge 0$, а значит, в области $M_{ij} \in [0,1]$ при $E_{ij} \in [0,1]$ и $K_{ij} \in [0,1]$ $\frac{\partial P(M)}{\partial M_{ij}} \ge 0$. Из этого следует, что в области допустимых значений ЦФ, заданная в виде (4.1), не имеет

локальных экстремумов, а следовательно, для ее максимизации может быть применен градиентный метод.

4.3. Разработка алгоритма решения задачи

Согласно [119, 120] решение задачи математического программирования градиентным методом заключается в том, что значение экстремума некоторой функции отыскивается путем последовательных шагов из начальной допустимой точки по направлению ее градиента (антиградиента), не выходя из области допустимых решений. Разные варианты градиентного метода заключаются в способе выбора шагового множителя на каждой итерации, тех или иных способах вычисления градиентов (аппроксимация, приближенные вычисления градиентов) и т. п.

Поскольку целью максимизации ЦФ (4.1) является поиск решения, оптимального по критерию затрат, предлагается при решении сформулированной задачи математического программирования в качестве направления движения из текущей допустимой точки использовать предложенное в [121, 122, 123] направление наибольшей удельной значимости по критерию затрат (отношения прироста показателя вероятности сохранения кибернетической безопасности к приросту стоимости), т. е. направление, соответствующее максимальному значению частной производной $\partial P(M_{ij})/\partial C(M_{ij})$, аппроксимированному отношением $\Delta P(M_{ij})/\Delta C(M_{ij})$, где $\Delta P(M_{ij})$ – приращение $P(M)$ при увеличении значения M_{ij} на величину шага, а $\Delta C(M_{ij})$ – приращение $C(M)$ при увеличении значения M_{ij} на величину шага. Поскольку M_{ij} может принимать только значения «0» или «1», величину шага предлагается выбрать равной «1».

Рассмотрим градиентный метод решения сформулированной задачи в следующей постановке:

$$P(M)=\prod_{i=1}^{L}P_i(M)= \\ =\prod_{i=1}^{L}\left(\prod_{j=1}^{N}(1-E_{ij})+\sum_{j=1}^{N}\left[E_{ij}\prod_{k=1}^{j-1}(1-E_{ik})\left(\sum_{k=1}^{j}M_{ik}K_{ik}\prod_{l=k+1}^{j}(1-M_{il}K_{il})\right)\right]\right)\to\max \tag{4.6}$$

$$C(M)=\sum_{i=1}^{L}\left(\sum_{j=1}^{N}M_{ij}\cdot C_{ij}\right)\le C_o, \tag{4.7}$$

$$M=\{M_{ij}\},\ \ M_{ij}\in\{0,1\}. \tag{4.8}$$

Пусть на начальном шаге процесса вычислений допустимое решение имеет компоненты $M_{ij} = 0$, $i \in I = \{1, \ldots, L\}$, $j \in J = \{1, \ldots, N\}$, т. е. отсутствуют механизмы защиты от всех угроз на всех уровнях стека протоколов.

На первом шаге среди всех кибернетических угроз $i \in I$ и для всех уровней стека протоколов $j \in J_i^{(0)}$, $J_i^{(0)} = J_{M_i}$, где J_{M_i} – множество номеров уровней стека протоколов, на которых применим выбранный механизм защиты от i-й кибернетической угрозы (определяется характеристиками конкретного механизма [11, 12, 13, 14, 15, 16, 17, 18, 19]), отыскиваются кибернетическая угроза $n \in I$ и уровень стека протоколов $m \in J_n^{(0)}$, такие, что реализация механизма защиты от данной угрозы на данном уровне стека протоколов (т. е. присвоение $M_{nm}^{(1)} = M_{nm}^{(0)} + 1$) приводит к наибольшему относительному приросту показателя вероятности сохранения кибернетической безопасности, т. е. наибольшему приросту на единицу стоимости. Поскольку, как показано в п. 2.1, в ИКС с открытой архитектурой имеется ранжирование значимостей механизмов защиты, реализованных на различных уровнях стека протоколов, т. е. механизм, реализованный на уровне m, обеспечивает защиту от кибернетических угроз, реализованных на уровнях стека протоколов, больших m, компоненты M_{ij} для $i = n$, $j \in \{m, \ldots, sup\, J_n^{(0)}\}$ принимаются равными нулю и исключаются из дальнейшего рассмотрения, т. е. $J_i^{(1)} = J_i^{(0)}$, $i \in I$, $i \neq n$, $J_n^{(1)} = J_n^{(0)} \setminus \{m, \ldots, sup\, J_n^{(0)}\}$.

На втором шаге отыскивается следующая кибернетическая угроза и для нее уровень стека протоколов, реализация механизма защиты на данном уровне стека протоколов приводит к наибольшему относительному приросту показателя вероятности сохранения кибернетической безопасности по отношению к стоимости. Этой новой угрозой может быть та же угроза, что и на первом шаге.

Пусть на s-м шаге процесса вычислений достигнуто допустимое решение $M^{(s)}$. Вероятность сохранения кибернетической безопасности определяется так:

$$P(M^{(s)}) = \prod_{i=1}^{L} P_i(M^{(s)}),$$

а стоимость реализации выбранного множества механизмов защиты как

$$C(M^{(s)}) = \sum_{i=1}^{L} \left(\sum_{j=1}^{N} M_{ij}^{(s)} C_{ij} \right).$$

На (s + 1)-м шаге для выбора направления, в котором функция вероятности возрастает максимально, определяем

$$\gamma^{(s+1)} = \max_{i \in I, j \in J_i^{(s)}} \tilde{\gamma}_{ji}^{(s)}(M^{(s)}),$$

где

$$\tilde{\gamma}_{ji}^{(s)}(M^{(s)}) = \frac{P(M^{(s)} \setminus M_{ij}^{(s)}, M_{ij}^{(s)} + 1) - P(M^{(s)})}{C(M^{(s)} \setminus M_{ij}^{(s)}, M_{ij}^{(s)} + 1) - C(M^{(s)})};$$

$$P(M^{(s)} \setminus M_{ij}^{(s)}, M_{ij}^{(s)} + 1) = \frac{P(M^{(s)}) P_i(M^{(s)} \setminus M_{ij}^{(s)}, M_{ij}^{(s)} + 1)}{P_i(M^{(s)})}.$$

В данном выражении

$$C(M^{(s)} \setminus M_{ij}^{(s)}, M_{ij}^{(s)} + 1) = C_{ij} + \sum_{i=1}^{L} \left(\sum_{j=1}^{N} M_{ij}^{(s)} C_{ij} \right) = C_{ij} + C(M^{(s)}).$$

Тогда

$$\tilde{\gamma}_{ji}^{(s)}(M_{ij}^{(s)}) = P(M^{(s)}) \frac{P_i(M^{(s)} \setminus M_{ij}^{(s)}, M_{ij}^{(s)} + 1) - P_i(M^{(s)})}{C_{ij} P_i(M^{(s)})} =$$

$$= P(M^{(s)}) \frac{\sum_{k=j}^{N} E_{ik} \prod_{l=1}^{k-1} (1 - E_{il}) \prod_{m=1}^{j-1} (1 - M_{im}^{(s)} K_{im}) \prod_{n=j+1}^{k} (1 - M_{in}^{(s)} K_{in})}{C_{ij} \left(\prod_{j=1}^{N} (1 - E_{ij}) + \sum_{j=1}^{N} \left[E_{ij} \prod_{k=1}^{j-1} (1 - E_{ik}) \left(\sum_{k=1}^{j} M_{ik}^{(s)} K_{ik} \prod_{l=k+1}^{j} (1 - M_{il}^{(s)} K_{il}) \right) \right] \right)}.$$

Таким образом,

$$\gamma^{(s+1)} = \max_{i \in I, j \in J_i^{(s)}} \tilde{\gamma}_{ji}^{(s)}(M_{ij}^{(s)}) =$$

$$= \max_{i \in I, j \in J_i^{(s)}} P(M^{(s)}) \frac{\sum_{k=j}^{N} E_{ik} \prod_{l=1}^{k-1} (1 - E_{il}) \prod_{m=1}^{j-1} (1 - M_{im}^{(s)} K_{im}) \prod_{n=j+1}^{k} (1 - M_{in}^{(s)} K_{in})}{C_{ij} \left(\prod_{j=1}^{N} (1 - E_{ij}) + \sum_{j=1}^{N} \left[E_{ij} \prod_{k=1}^{j-1} (1 - E_{ik}) \left(\sum_{k=1}^{j} M_{ik}^{(s)} K_{ik} \prod_{l=k+1}^{j} (1 - M_{il}^{(s)} K_{il}) \right) \right] \right)} =$$

$$= P(M^{(s)}) \max_{i \in I, j \in J_i^{(s)}} \frac{\sum_{k=j}^{N} E_{ik} \prod_{l=1}^{k-1} (1 - E_{il}) \prod_{m=1}^{j-1} (1 - M_{im}^{(s)} K_{im}) \prod_{n=j+1}^{k} (1 - M_{in}^{(s)} K_{im})}{C_{ij} \left(\prod_{j=1}^{N} (1 - E_{ij}) + \sum_{j=1}^{N} \left[E_{ij} \prod_{k=1}^{j-1} (1 - E_{ik}) \left(\sum_{k=1}^{j} M_{ik}^{(s)} K_{ik} \prod_{l=k+1}^{j} (1 - M_{il}^{(s)} K_{il}) \right) \right] \right)} =$$

$$= P(M^{(s)}) \max_{i \in I, j \in J_i^{(s)}} \gamma_{ji}^{(s)}.$$

где

$$\gamma_{ji}^{(s)} = \frac{\sum_{k=j}^{N} E_{ik} \prod_{l=1}^{k-1}(1-E_{il}) \prod_{m=1}^{j-1}(1-M_{im}^{(s)}K_{im}) \prod_{n=j+1}^{k}(1-M_{in}^{(s)}K_{in})}{C_{ij}\left(\prod_{j=1}^{N}(1-E_{ij}) + \sum_{j=1}^{N}\left[E_{ij}\prod_{k=1}^{j-1}(1-E_{ik})\left(\sum_{k=1}^{j} M_{ik}^{(s)}K_{ik}\prod_{l=k+1}^{j}(1-M_{il}^{(s)}K_{il})\right)\right]\right)}.$$

Поскольку $P(M^{(s)})$ входит сомножителем во все величины $\tilde{\gamma}_{ji}^{(s)}(M_{ij}^{(s)})$ и не влияет на выбор направления движения, то можно упростить вычисления. Для этого на шаге $(s+1)$ необходимо двигаться в направлении $\gamma^{(s+1)} = \max\limits_{i\in I, j\in J_i^{(s)}} \gamma_{ji}^{(s)}(M_{ij}^{(s)})$. Выигрыш от этой модификации очевиден.

Кроме этого, для учета ранжирования показателей значимости механизмов защиты, реализованных на разных уровнях стека протоколов, предлагается выполнить модификацию классического градиентного метода. Модификация метода должна заключаться в следующем: если на очередной итерации в допустимое решение $M^{(s+1)}$ включен механизм $M_{nm}{}^{(s+1)}$, реализованный на более низком уровне стека протоколов m, то из допустимого решения $M^{(s+1)}$ должны быть исключены механизмы, реализованные на более высоких уровнях стека протоколов, т. е. $M_{nj}{}^{(s+1)}=0, j \in \{m+1, \ldots, sup\ J_n^{(s)}\}$. Дальнейшая оптимизация по обнуленным компонентам $M^{(s+1)}$ проводиться не должна.

Поиск оптимального решения должен завершаться либо при невозможности выполнения на $(s+1)$-м шаге ограничения $C(M^{(s+1)}) \le C_0$, либо в случае, когда из рассмотрения исключены все $M_{ij}, i \in I, j \in J$, т. е. $J_i^{(s)} = \varnothing, i \in I$, либо в случае, когда $\gamma^{(s+1)} = 0$, что означает достижение глобального максимума.

Таким образом, предлагаемый алгоритм решения задачи (4.6) – (4.8) состоит в следующем.

Шаг 0. Полагаем $s=0$, $I=\{1,...,L\}$, $J=\{1,...,N\}$, $M_{ij}=0,\ i\in I,\ j\in I$. На основании характеристик выбранных механизмов определяем множество номеров уровней стека протоколов, на которых они применимы $J_{M_i},\ i\in I$, задаем $J_i^{(s)} = J_{M_i},\ i\in I$ и переходим к шагу 1.

Шаг 1. Если

$$J_i^{(s)} = \varnothing,\ i \in I,$$

то заканчиваем вычисления, $M^{(s)}$ – оптимальное решение.

Шаг 2. Вычисляем

$$\gamma_{ji}^{(s)} = \frac{\sum_{k=j}^{N} E_{ik} \prod_{l=1}^{k-1}(1-E_{il}) \prod_{m=1}^{j-1}(1-M_{im}^{(s)}K_{im}) \prod_{n=j+1}^{k}(1-M_{in}^{(s)}K_{in})}{C_{ij}\left(\prod_{j=1}^{N}(1-E_{ij}) + \sum_{j=1}^{N}\left[E_{ij}\prod_{k=1}^{j-1}(1-E_{ik})\left(\sum_{k=1}^{j} M_{ik}^{(s)}K_{ik}\prod_{l=k+1}^{j}(1-M_{il}^{(s)}K_{il})\right)\right]\right)}, i \in I, j \in J^{(s)}$$

и определяем

$$\gamma^{(s+1)} = \gamma_{nm}^{(s)}(M_{nm}^{(s)}) = \max_{i \in I,\, j \in J_i^{(s)}} \gamma_{ji}^{(s)}(M_{ij}^{(s)}).$$

Шаг 3. Если

$$\gamma_{ij}^{(s+1)} = 0,$$

то заканчиваем вычисления, $M^{(s)}$ – оптимальное решение.

Шаг 4. Полагаем

$$M_{nm}^{(s+1)} = M_{nm}^{(s)}+1;$$
$$M_{ij}^{(s+1)} = M_{ij}^{(s)},\ i \in I,\ i \neq n, j \in J;$$
$$M_{nj}^{(s+1)} = M_{nj}^{(s)}, j \in \{1, \ldots, m-1\};$$
$$M_{nj}^{(s+1)} = 0, j \in \{m+1, \ldots, sup\, J_n^{(s)}\};$$
$$J_i^{(s+1)} = J_i^{(s)},\ i \in I,\ i \neq n;$$
$$J_n^{(s+1)} = J_n^{(s)} \setminus \{m, \ldots, sup\, J_n^{(s)}\},$$

и переходим к шагу 5.

Шаг 5. Если $C(M^{(s)}) \leq C_0 < C(M^{(s+1)})$, то заканчиваем вычисления, $M^{(s)}$ – оптимальное решение. В противном случае полагаем $s = s + 1$ и переходим к шагу 1.

Шаг 6. Окончание алгоритма.

Полученное в результате выполнения данной процедуры значение $M^{(s)}$ определяет набор механизмов защиты, который необходимо реализовать в ИКС для обеспечения максимального уровня кибернетической безопасности для множества выявленных кибернетических угроз T с показателями эффективности $E = \{E_{ij}\}$, $i \in I, j \in J$ при выполнении заданного ограничения C_o на общую стоимость средств защиты.

В разработанном алгоритме за счет учета вида ЦФ (4.1) и ограничений (4.2) удалось существенно уменьшить объем вычислений, необходимых для определения значений γ_{ij}. Кроме этого, за счет модификации классического градиентного метода и учета на шаге 4 алгоритма ранжирования значимостей механизмов защиты информации, реализованных на различных уровнях стека протоколов, удалось

избежать дублирования механизмами защиты, реализованными на более высоких уровнях стека протоколов, механизмов, реализованных на более низких уровнях (с исключением соответствующих затрат на реализацию механизмов защиты).

4.4. Исследование разработанного алгоритма

Для проверки сходимости предложенного алгоритма была разработана его программная реализация и проведено численное исследование на различных тестовых примерах. Исходные данные тестовых примеров выбирались таким образом, чтобы можно было убедиться в сходимости алгоритма при различных закономерностях изменения показателей эффективности реализации кибернетических угроз E_{ij}, при различных закономерностях распределения кибернетических угроз и применимости механизмов защиты по уровням стека протоколов, при различных закономерностях изменения стоимостей реализации механизмов защиты информации C_{ij}.

В качестве примера 1 использовалась задача построения оптимальной СЗИ для обеспечения кибернетической безопасности ИКС с 7 уровнями стека протоколов для 20 кибернетических угроз при следующих исходных данных:

– попытки реализации кибернетических угроз распределены по уровням стека протоколов случайным образом (закон распределения равномерный), каждая угроза может быть реализована на одном уровне стека протоколов;

– значения показателей эффективности реализации кибернетических угроз одинаковы и не зависят от уровня стека протоколов;

– применимости механизмов защиты распределены по уровням стека протоколов по тому же закону, что и кибернетические угрозы;

– стоимость реализации механизмов защиты обратно пропорциональна уровню стека протоколов, на котором применим данный механизм;

– предельная стоимость реализации СЗИ равна 50 % от общей полной стоимости потенциально применимых средств защиты (2776,19 ед.).

В качестве примера 2 использовалась задача построения оптимальной СЗИ для обеспечения кибернетической безопасности ИКС с 7 уровнями стека протоколов для 20 кибернетических угроз при следующих исходных данных:

– попытки реализации кибернетических угроз распределены по уровням стека протоколов случайным образом (закон распределения равномерный), каждая угроза может быть реализована на одном уровне стека протоколов;

– значения показателей эффективности реализации кибернетических угроз пропорциональны квадрату уровня стека протоколов;

– применимости механизмов защиты распределены по уровням стека протоколов по тому же закону, что и угрозы информации;

– стоимость реализации механизмов защиты обратно пропорциональна уровню стека протоколов, на котором применим данный механизм;

– предельная стоимость реализации СЗИ равна 50 % от общей полной стоимости потенциально применимых средств защиты (2776,19 ед.).

В качестве примера 3 использовалась задача построения оптимальной СЗИ для обеспечения кибернетической безопасности ИКС с 7 уровнями стека протоколов для 20 кибернетических угроз при следующих исходных данных:

– попытки реализации кибернетических угроз распределены по уровням стека протоколов случайным образом (закон распределения равномерный), каждая угроза может быть реализована на одном уровне стека протоколов;

– значения показателей эффективности реализации кибернетических угроз пропорциональны квадрату уровня стека протоколов;

– механизмы защиты применимы на всех уровнях стека протоколов;

– стоимость реализации механизмов защиты обратно пропорциональна уровню стека протоколов, на котором применим данный механизм;

– предельная стоимость реализации СЗИ равна 10 % от общей полной стоимости потенциально применимых средств защиты (5185,71 ед.).

В качестве примера 4 использовалась задача построения оптимальной СЗИ для обеспечения кибернетической безопасности ИКС с 7 уровнями стека протоколов для 20 кибернетических угроз при следующих исходных данных:

– попытки реализации кибернетических угроз распределены по уровням стека протоколов случайным образом (закон распределения равномерный), каждая угроза может быть реализована на трех уровнях стека протоколов;

– значения показателей эффективности реализации угроз пропорциональны квадрату уровня стека протоколов;

– применимости механизмов защиты распределены по уровням стека протоколов случайным образом (закон распределения равномерный), каждый механизм может быть применен на трех уровнях стека протоколов;

– стоимость реализации механизмов защиты обратно пропорциональна уровню стека протоколов, на котором применим данный механизм;

– предельная стоимость реализации СЗИ равна 30 % от общей полной стоимости потенциально применимых средств защиты (6200,00 ед.).

В качестве примера 5 использовалась задача построения оптимальной СЗИ для обеспечения кибернетической безопасности ИКС с 7 уровнями стека протоколов для 20 кибернетических угроз при следующих исходных данных:

– попытки реализации кибернетических угроз распределены по уровням стека протоколов случайным образом (закон распределения равномерный), каждая угроза может быть реализована на трех уровнях стека протоколов;

– значения показателей эффективности реализации угроз пропорциональны квадрату уровня стека протоколов;

– механизмы защиты применимы на всех уровнях стека протоколов;

– стоимость реализации механизмов защиты обратно пропорциональна уровню стека протоколов, на котором применим данный механизм;

– предельная стоимость реализации СЗИ равна 10 % от общей полной стоимости потенциально применимых средств защиты (5185,71 ед.).

Исследование проводилось для двух вариантов реализации алгоритма: с использованием немодифицированного и модифицированного с учетом ранжирования значимостей градиентного метода. При этом значения всех коэффициентов эффективности механизмов защиты принимались равными единице.

Для исходных данных примера 1 оптимизационная процедура на основе немодифицированного градиентного метода завершилась на 14-й итерации, при этом было выполнено 189 вычислений градиента ЦФ и достигнуто значение $P(M) = 0{,}735092$ при стоимости средств защиты, равной 2719,05 ед., что составляет 97,9 % от максимально допустимой. Аналогичная процедура на основе модифицированного градиентного метода также завершилась на 14-й итерации, при этом также было выполнено 189 вычислений градиента ЦФ и достигнуто значение $P(M) = 0{,}735092$ при том же значении стоимости средств защиты.

Для исходных данных примера 2 оптимизационная процедура на основе немодифицированного градиентного метода завершилась на 14-й итерации, при этом было выполнено 189 вычислений градиента ЦФ и достигнуто значение $P(M) = 0{,}913618$ при стоимости средств защиты, равной 2719,05 ед., что составляет 97,9 % от максимально допустимой. Аналогичная процедура на основе модифицированного градиентного метода также завершилась на

14-й итерации, при этом было выполнено 189 вычислений градиента ЦФ и достигнуто значение $P(M)$ = 0,913618 при том же значении стоимости средств защиты.

Для исходных данных примера 3 оптимизационная процедура на основе немодифицированного градиентного метода завершилась на 19-й итерации, при этом было выполнено 2489 вычислений градиента ЦФ и достигнуто значение $P(M)$ = 0,997817 при стоимости средств защиты, равной 4552,38 ед., что составляет 87,8 % от максимально допустимой. Аналогичная процедура на основе модифицированного градиентного метода также завершилась на 19-й итерации, при этом было выполнено 2241 вычисление градиента ЦФ и достигнуто значение $P(M)$ = 0,997817 при том же значении стоимости средств защиты.

Для исходных данных примера 4 оптимизационная процедура на основе немодифицированного градиентного метода завершилась на 23-й итерации, при этом было выполнено 1058 вычислений градиента ЦФ и достигнуто значение $P(M)$ = 0,935358 при стоимости средств защиты, равной 5850 ед., что составляет 94,4 % от максимально допустимой. Аналогичная процедура на основе модифицированного градиентного метода завершилась на 26-й итерации, при этом было выполнено 974 вычисления градиента ЦФ и достигнуто значение $P(M)$ = 0,959804 при стоимости средств защиты, равной 6166,67 ед., что составляет 99,5 % от максимально допустимой.

Для исходных данных примера 5 оптимизационная процедура на основе немодифицированного градиентного метода завершилась на 23-й итерации, при этом было выполнено 2967 вычислений градиента ЦФ и достигнуто значение $P(M)$ = 0,901002 при стоимости средств защиты, равной 5102,38 ед., что составляет 98,4 % от максимально допустимой. Аналогичная процедура на основе модифицированного градиентного метода завершилась на 28-й итерации, при этом было выполнено 2950 вычислений градиента ЦФ и достигнуто значение $P(M)$ = 0,959804 при стоимости средств защиты, равной 5142,86 ед., что составляет 99,2 % от максимально допустимой.

Результаты численного исследования предложенного алгоритма для исходных данных примеров 1–5 с использованием немодифицированного градиентного метода приведены на рис. 4.1–4.3. Результаты численного исследования предложенного алгоритма для исходных данных примеров 1–5 с использованием модифицированного градиентного метода приведены на рис. 4.4–4.6. Результаты

сравнения вычислительной эффективности алгоритмов на основе немодифицированного и модифицированного методов приведены на рис. 4.7 и 4.8.

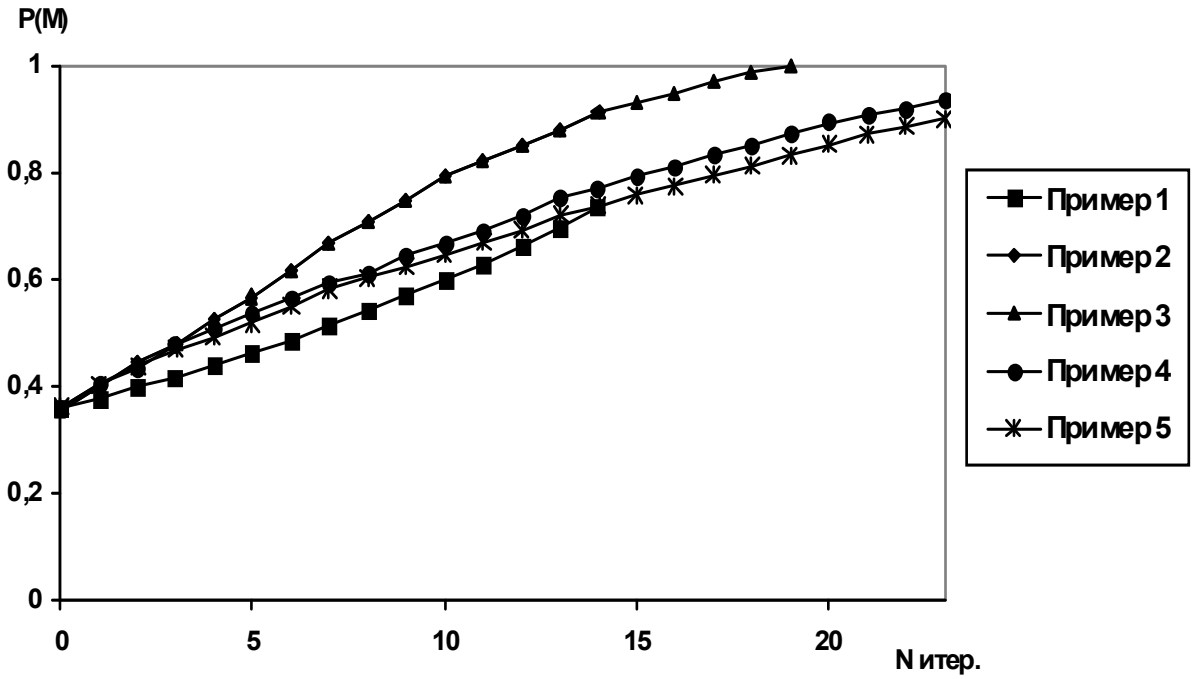


Рис. 4.1. Изменение значения $P(M)$ в ходе градиентной процедуры (немодифицированный градиентный метод)

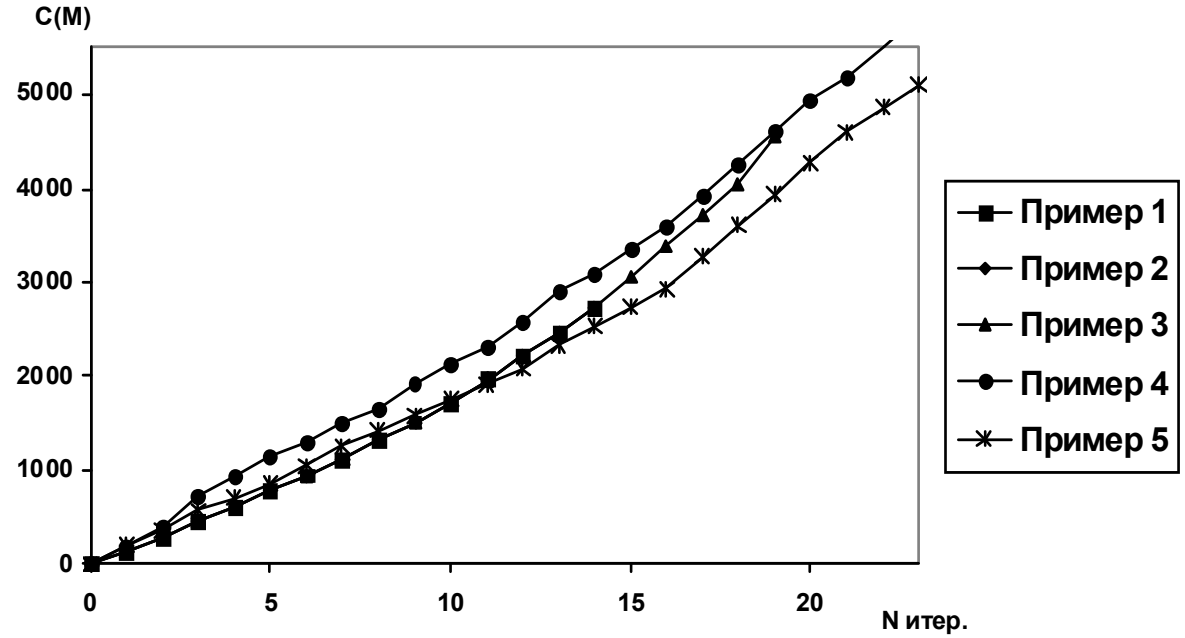


Рис. 4.2. Изменение значения $C(M)$ в ходе градиентной процедуры (немодифицированный градиентный метод)

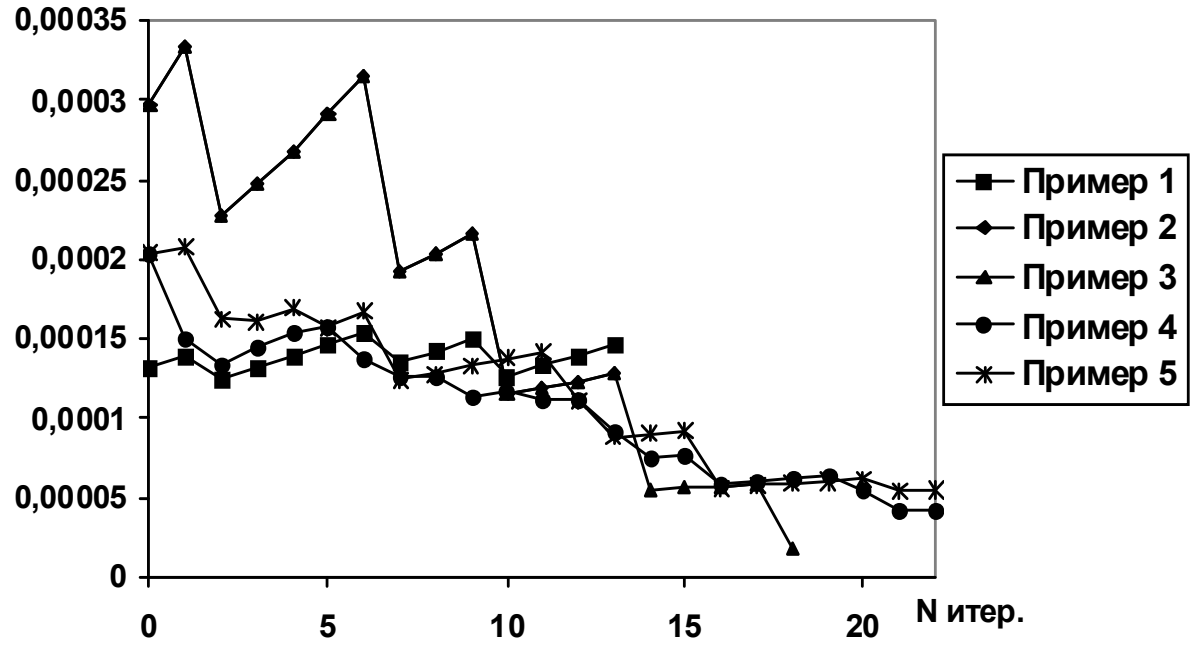


Рис. 4.3. Изменение значения $\Delta P(M)/\Delta C(M)$ в ходе градиентной процедуры (немодифицированный градиентный метод)

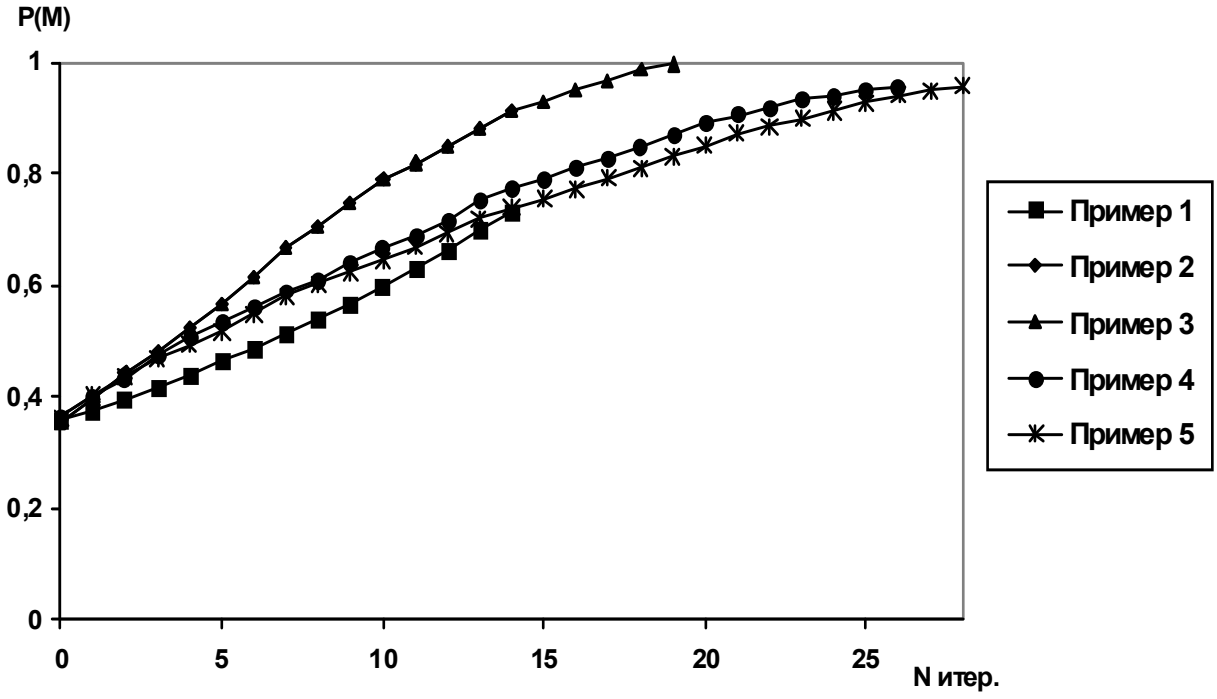


Рис. 4.4. Изменение значения $P(M)$ в ходе градиентной процедуры (модифицированный градиентный метод)

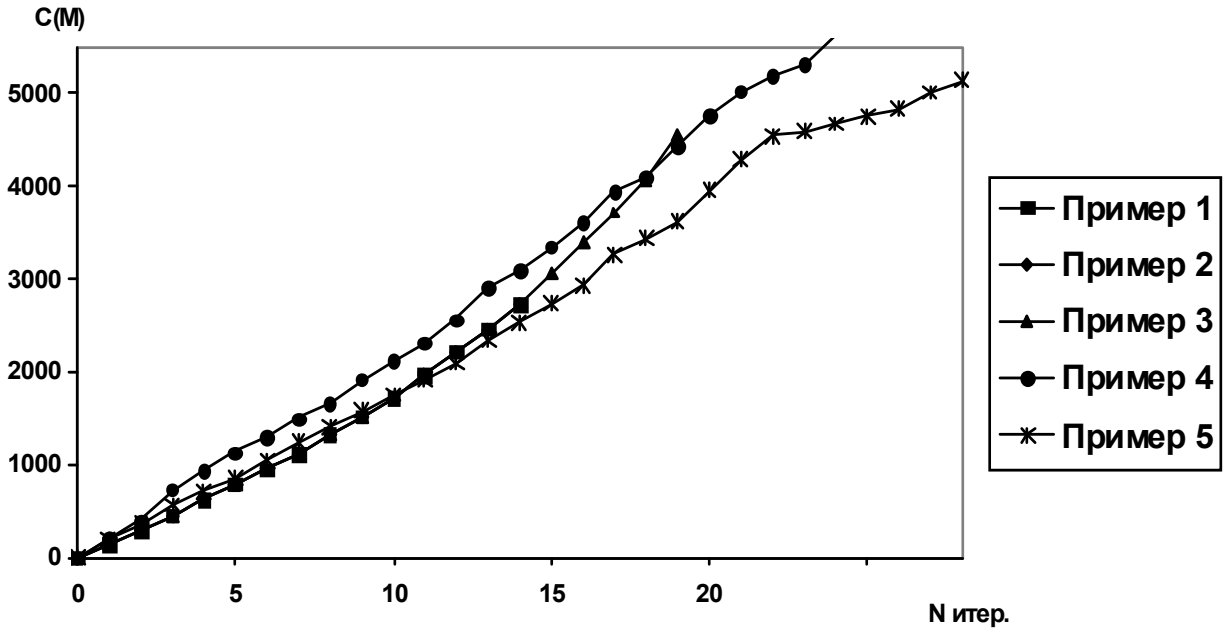


Рис. 4.5. Изменение значения $C(M)$ в ходе градиентной процедуры (модифицированный градиентный метод)

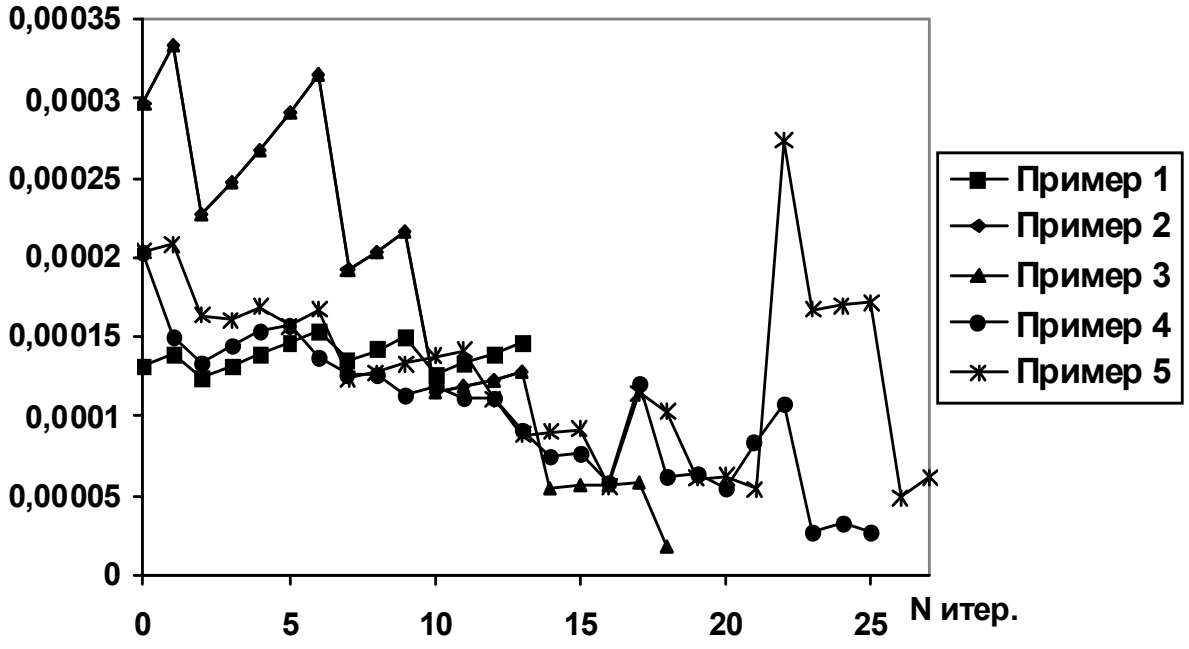


Рис. 4.6. Изменение значения $\Delta P(M)/\Delta C(M)$ в ходе градиентной процедуры (модифицированный градиентный метод)

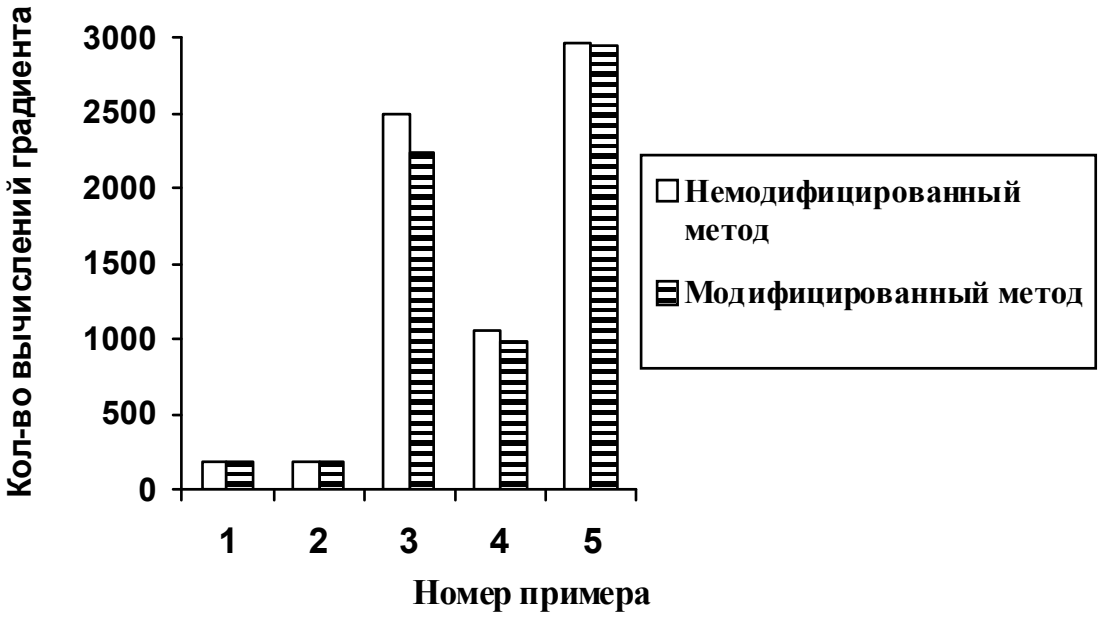


Рис. 4.7. Результаты сравнения вычислительной эффективности алгоритмов на основе немодифицированного и модифицированного градиентных методов

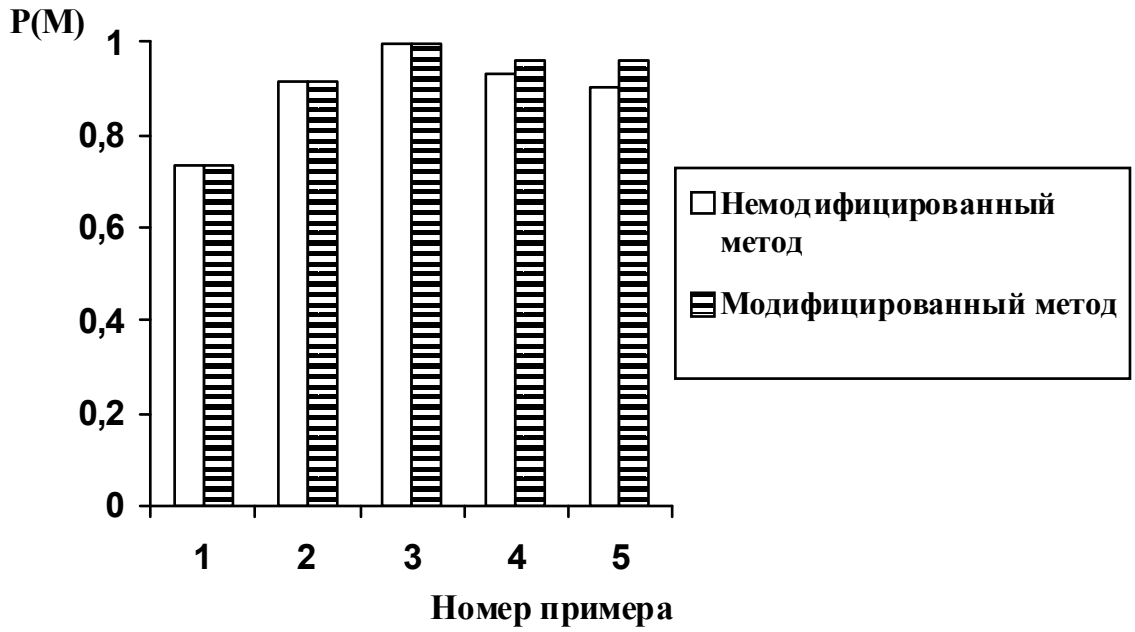


Рис. 4.8. Достигнутые значения $P(M)$ для немодифицированного и модифицированного методов

Как видно из анализа приведенных на рис. 4.1 и 4.4 графиков, изменения значения $P(M)$ в ходе градиентной процедуры на каждом очередном шаге алгоритма, независимо от варианта исходных данных и используемого градиентного

метода, достигается большее значение функции вероятности сохранения кибернетической безопасности $P(M)$ за счет использования новых механизмов защиты и роста их суммарной стоимости $C(M)$, что видно из рис. 4.2 и 4.5. При этом аппроксимированное значение градиента $\Delta P(M)/\Delta C(M)$, как видно из рис. 4.3 и 4.6, уменьшается, что подтверждает факт приближения к абсолютному максимуму $P(M)$. Сравнение результатов, полученных с использованием немодифицированного и модифицированного градиентных методов, позволяет сделать вывод о том, что при тех же исходных данных алгоритм с использованием модифицированного градиентного метода, во-первых, требует меньших вычислительных затрат (рис. 4.7), уменьшение объема вычислений достигает 10 %, а во-вторых, обладает большей эффективностью, так как полученное решение (рис. 4.8) обеспечивает увеличение достигнутого уровня кибернетической безопасности до 6 %.

Выводы

Для решения задачи построения оптимальной СЗИ с учетом ограничений на стоимость средств защиты сформулирована соответствующая задача математического программирования, в которой решаемая оптимизационная задача рассматривается как задача нелинейного целочисленного программирования с одним ограничением. Обоснована возможность решения указанной задачи с использованием градиентного метода.

Разработан алгоритм решения сформулированной задачи на основе модифицированного градиентного метода, в котором на каждой итерации в качестве направления движения принимается направление, соответствующее наибольшей удельной значимости по критерию затрат, и для учета закономерностей функционирования механизмов защиты в ИКС с открытой архитектурой выполняется соответствующая корректировка вектора изменяемых параметров.

Проведено вычислительное исследование разработанного алгоритма, результаты которого подтвердили его устойчивую сходимость к оптимальному решению, а также преимущества использования модифицированного градиентного метода по сравнению с обычным: снижение объема вычислений более чем на 10 % и нахождение решений, обеспечивающих повышение уровня кибернетической безопасности более чем на 6 %.

5. СТРУКТУРНАЯ ЛОГИКО-ВЕРОЯТНОСТНАЯ МОДЕЛЬ ОЦЕНКИ КОМПЛЕКСНОГО УРОВНЯ ЗАЩИЩЕННОСТИ ИКС

В данной главе приводится модель ИКС с учетом топологии сети, строится комплексная модель киберугроз на ИКС, основанная на графе атак, и модель системы защиты. Для оценки уровня защищенности ИКС применяется логико-вероятностный метод анализа безопасности ССС и строится функция вероятности успешной кибератаки. Далее описывается методика оценки уровня защищенности ИКС.

5.1. Построение комплексной модели киберугроз на ИКС

Для построения комплексной модели киберугроз сначала построим модель ИКС, которая учитывает топологию сети и на основе которой можно определить зависимости между объектами системы.

5.1.1. Модель ИКС

Для построения модели ИКС в качестве основы возьмем представление логической топологии сети в виде ориентированного графа $G(V,E)$, где $v_i \in V$ – множество объектов/информационных ресурсов/сервисов системы; $E = \{0,1\}$ – наличие или отсутствие связей между ними, $E \subseteq V \times V$ и $e_i \in E$.

Топология сети ИКС будет зависеть от коммутационных соединений и соответствующих сетевых настроек. При построении логической топологии корпоративной сети следует также брать во внимание возможные информационные потоки, с учетом которых можно более точно в дальнейшем выполнять построение сценариев атак.

Если на одном физическом сервере располагается несколько сервисов, которые могут быть объектом или источником атаки или через них может проходить сценарий кибератаки, то будем выделять их в отдельные объекты – v_i.

Полученный граф $G(V,E)$ представлен в виде матрицы, которая названа матрицей доступности объектов. Граф является ориентированным, потому что с некоторых объектов можно инициировать соединение только в одностороннем порядке (например, от АРМ администратора к соответствующему сервису системы управления базами данных). При одноранговой (полностью

связной) топологии сети каждый из объектов имеет доступ ко всем другим объектам и матрица доступности будет заполнена «1». В случае, когда все объекты не будут иметь связей между собой, матрица доступности будет заполнена «0», кроме главной диагонали.

Для построения матрицы доступности сервисов уже существующей ИКС лучше всего использовать сканеры сети, такие как Nmap. Запуская их на каждом из хостов, можно получить реальную картину относительно доступных хостов и сервисов, которые на них работают. Подобный подход также используется при построении графа атак.

Таким образом, на основе матрицы доступности можно определить «маршруты» между объектами v_0 и v_t в ИКС, которые будут иметь следующий вид:

$$S_k = (v_0, e_1, v_1, ..., v_{t-1}, e_t, v_t) = (v_0; v_t), \quad (5.1)$$

где $v_i \in V$, $e_i \in E$.

Учитывая, что между некоторыми объектами может быть несколько маршрутов, будем обозначать их следующим образом:

$$S = \{S_1, ..., S_p\} = \{S_1 = (v_0; v_t), ..., S_p = (v_0; v_t)\} = \{(v_0; v_t)\}, \quad (5.2)$$

где p – количество разных маршрутов.

При построении модели ИКС будем также учитывать, что топология современной корпоративной сети обычно состоит из нескольких сегментов (или подсетей), взаимодействие и передача информации между которыми обеспечиваются при помощи настройки маршрутизаторов (или средств, которые поддерживают маршрутизацию). В пределах этих сегментов каждый из объектов имеет доступ ко всем другим объектам этого же сегмента.

Таким образом, матрица доступности объектов будет иметь блочно-диагональный вид, представленный на рис. 5.1.

	1	2	3	...	$N-2$	$N-1$	N
1	1	1	1	...	0	0	0
2	1	1	1	...	0	0	0
3	1	1	1	...	0	0	0
...	...	...	...	...	...	...	...
$N-2$	0	0	0	...	1	1	1
$N-1$	0	0	0	...	1	1	1
N	0	0	0	...	1	1	1

$$\begin{pmatrix} S_1 & 0 & & 0 & 0 \\ 0 & S_2 & & & \\ & & & & \\ 0 & & & \ldots & 0 \\ & & & & \\ 0 & & & & S_T \end{pmatrix}$$

Рис. 5.1. Матрица доступности объектов ИКС

В общем случае при количестве сегментов, равном T, матрица доступности объектов представляется в форме, приведенной на рис. 5.1, где S_i – сегмент сети, который представляет собой матрицу, заполненную «1». Каждый из объектов ИКС $V = \{v_1,...,v_N\}$ будет принадлежать одному из сегментов $v_j \in S_k$, это обозначается как пара $s_{jk} = (v_j \mid S_k)$, где $k \in [1,T]$. За настройки маршрутизаторов отвечают значения, находящиеся за границами блоков S_i. Тогда множество пар $\{s_{jk}\}, j = \overline{1,N}$ будет определять модель логической топологии сети:

$$G = \left(s_{jk} = (v_j \mid S_k) \mid k \in [1,T]; j = \overline{1,N}\right). \quad (5.3)$$

Более детально модель ИКС с учетом сегментации сети, а также принципы сегментации сети будут рассмотрены в следующей главе.

5.1.2. Комплексная модель киберугроз ИКС

Сначала определим, что будет пониматься под успешной кибератакой на ИКС, или условия, при которых система будет переходить в опасное состояние.

Для ИКС условием перехода системы в опасное состояние будет являться успешная кибератака хотя бы на один из ее критических объектов. Под успешной кибератакой понимается нарушение политики безопасности информации и/или нанесение существенного ущерба [4] для одного из критических объектов киберсистемы.

Далее необходимо проанализировать способы осуществления данных атак и условия их проведения. Такой анализ будет представлять собой сценарий кибератаки.

Кроме этого, во время определения уровня защищенности ИКС также учтем ее топологию, размещение и информационную зависимость сервисов друг от друга. Исходя из этого, сценарий кибератаки будем формировать как последовательность действий злоумышленника, которые ему необходимо выполнить для успешной атаки на определенный объект системы – цель кибератаки. Такие действия злоумышленника будут состоять из последовательного захвата объектов, которые связаны информационными потоками с целевым объектом, начиная с того сервиса/объекта, к которому он имеет или может получить доступ. Данный сервис/объект как раз будет принадлежать поверхности атаки.

Под захватом объекта (сервиса, ресурса) понимается возможность злоумышленника выполнять с помощью этого объекта (сервиса) следующие действия:

– возможность захватить связанный с ним объект (сервис);

– атаковать критический объект (сервис) системы – цель кибератаки.

Таким образом, для построения сценария атак необходимо знать связь (топологию) сервисов ИКС между собой, а также источники кибератаки и цели (объекты) кибератаки.

Для построения комплексной модели киберугроз ИКС среди ее объектов $V = \{v_1,...,v_N\}$ выделяется множество критических объектов, которые считаются объектами кибератаки $O = (o_1,...,o_M) \subset V$, и множество объектов, которые считаются источниками кибератаки $A = (a_1,...,a_K) \subset V$.

Условием успешности кибератаки на ИКС считается успешная атака хотя бы на один из ее критических объектов: $O = (o_1,...,o_M) \subset V$.

К числу объектов кибератаки могут быть отнесены с точки зрения безопасности и функционирования сервисы и серверы ИКС, а к числу источников кибератаки – компьютеры пользователей, сервисы, подключенные к Интернет и внешним сетям (т. е. множество, входящее в поверхность атаки).

Множество объектов кибератаки (критических объектов) системы может состоять из сервисов, серверов, сетевого оборудования и других критических компонентов ИКС. Важным является то, что при попытке увеличить защищенность одного из компонентов защищенность другого может уменьшаться.

Кортеж $G(V,E,A,O)$, который включает в себя граф сети $G(V,E)$, объекты $O \subset V$ и источники кибератаки $A \subset V$, назовем графом кибератаки. На основе данных, которые содержит граф кибератаки (объекты O и источники A, а также топология сети $G(V,E)$), для каждого объекта атаки $o_t \in O$ строятся сценарии атаки в соответствии с (5.1), которые представляют собой все возможные (для данной топологии сети) пути от источников атаки A к объекту атак o_t:

$$K_{\phi_l} = R_l = (a_l, e_l, v_l, ..., v_{t-1}, e_t, o_t) = (a_l; o_t), \qquad (5.4)$$

где R_l – некоторый из маршрутов на графе между объектами a_l и o_t.

Согласно выражению (5.2) от каждого источника атак a_l к объекту атаки o_t может существовать от нуля до p путей в зависимости от топологии сети:

$$0 \le \left|K_{\phi}\right| \le p.$$

Набор путей от источников кибератаки $A = (a_1,...,a_K)$ к выбранному объекту атаки o_t будет содержать множество всех возможных путей проникновения злоумышленника – сценарии кибератак:

$$(A;o_t) = \{\{(a_1;o_t)\},\{(a_2;o_t)\},...,\{(a_K;o_t)\}\}. \quad (5.5)$$

Для наглядности представления и удобства анализа $G(V,E,A,O)$ можно представить в виде графа или дерева атак, где объекты, которые входят в один сценарий кибератаки, объединяются с помощью логического символа «И», а разные сценарии кибератак объединяются с помощью логического символа «ИЛИ», как это показано на рис. 5.2.

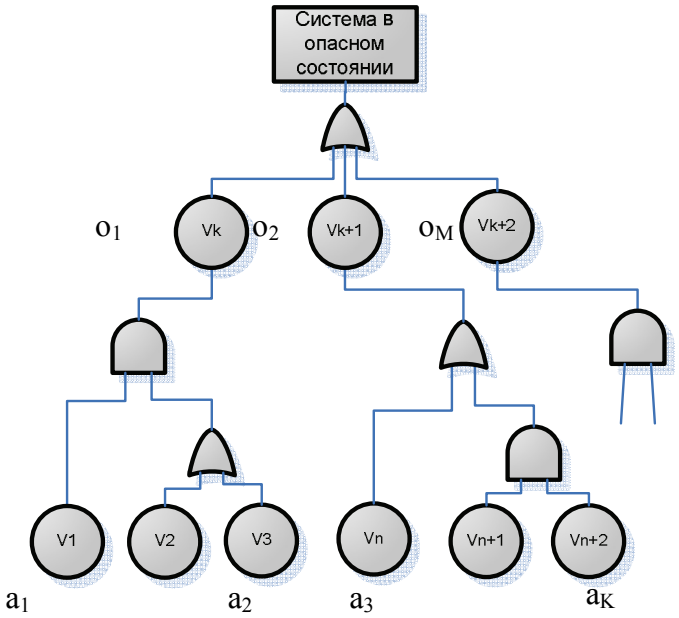


Рис. 5.2. Граф кибератаки: $A = (a_1,a_2,...,a_K) = (v_1,v_3,v_{n+2})$ – вершины-источники атак;

$O = (o_1,o_2,...,o_M) = (v_k,v_{k+1},v_{k+2})$ – вершины-объекты атак

Возможность злоумышленника захватить промежуточные объекты и успешно атаковать цель атаки будем определять как оценочные значения вероятностей $\{P_1,...,P_N\}$ для каждого объекта $V = \{v_1,...,v_N\}$ ИКС. Данные вероятности захвата объектов независимы и являются соответствующей характеристикой объектов ИКС. Вероятности захвата объектов подробнее рассмотрим в следующем пункте.

Таким образом, комплексную модель киберугроз будет определять следующий кортеж, который можно представить в виде графа кибератаки:

$$G(V,E,A,O,P), \quad (5.6)$$

где $G(V,E)$ – топология сети ИКС;

$A \subset V$ – множество источников кибератаки;

$O \subset V$ – множество объектов кибератаки;

$P = \{P_1,...,P_N\}$ – вероятности захвата объектов ИКС.

5.1.3. Оценка количества сценариев кибератак

Для дальнейшей оценки сложности алгоритмов определим приблизительное количество разных сценариев кибератак, которые могут быть построены на основе графа кибератак $G(V,E,A,O)$ для ИКС. При этом будем учитывать такую особенность – если один маршрут сценария атаки является частью другого, то из них останется только тот маршрут, который короче, т. е., если имеем маршруты на графе S_1 и S_2, и $S_1 \subset S_2$, то останется только маршрут S_1. Это следует из того, что этот маршрут является самым коротким сценарием совершения атаки.

В гл. 1 было показано, что общая архитектура современных информационно-коммуникационных систем является многоуровневой, т. е. при доступе к данным или конечным сервисам клиентские запросы заранее проходят через несколько промежуточных сервисов. Допустим, что при такой архитектуре все клиенты будут источниками атак, а конечные сервисы – объектами атак.

Максимальным количество разных сценариев атак будет тогда, когда объект верхнего уровня будет соединен со всеми объектами нижнего уровня, как это показано на рис. 5.3.

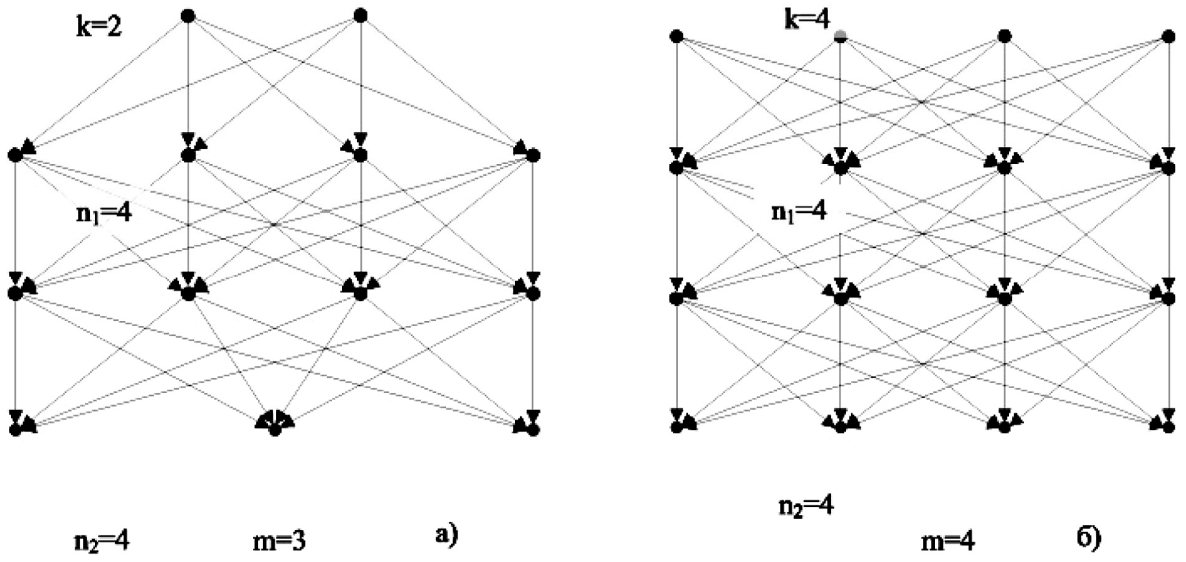


Рис. 5.3. Варианты сценариев кибератак

Общее количество сценариев, согласно (5.4)–(5.5), будет определяться количеством объектов на всех уровнях по формуле

$$\left|K_{\phi}\right| = k \cdot n_1 \cdot n_2 \cdot \ldots \cdot n_p \cdot m\,,$$

где k – количество источников кибератак;

m – количество объектов кибератаки;

$n_1 \ldots\ldots n_p$ – количество объектов на промежуточных уровнях, где p – количество уровней.

Для примера на рис. 5.3, *а* и *б* на верхнем уровне находятся источники атак (k), а на нижнем уровне – объекты атак (m), между ними находятся промежуточные сервисы (n_1) и (n_2).

Таким образом, для схемы на рис. 5.3, *а*, которая содержит 13 объектов, будем иметь следующее число возможных сценариев атак:

$$\left|K_{\phi_a}\right| = k \cdot n_1 \cdot n_2 \cdot m = 2 * 4 * 4 * 3 = 96\,,$$

а для схемы на рис. 5.3, *б*, которая содержит 16 объектов, будем иметь

$$\left|K_{\phi_a}\right| = k \cdot n_1 \cdot n_2 \cdot m = 4 * 4 * 4 * 4 = 256\,.$$

Исходя из того, что общая архитектура корпоративных приложений является трех- или четырехуровневой, будем считать, что количество промежуточных уровней будет равно четырем (p = 4). Тогда приблизительное максимальное количество сценариев атак для ИКС, которая содержит N объектов, будет

$$\left|K_{\phi_{\max}}\right| = \left(\frac{N}{4}\right)^4.$$

Например, если система содержит 100 объектов (N =100), то максимальное количество возможных сценариев атак $\left|K_{\phi_{\max}}\right|$ = 390625.

Из этого можно сделать вывод о том, что наибольшее количество сценариев кибератак будет в случае полносвязной сети, и, соответственно, для ее защиты потребуется гораздо больше усилий и ресурсов.

5.2. Определение вероятностей захвата объектов ИКС

Возможность злоумышленника захватить промежуточные объекты и успешно атаковать цель атаки характеризуется вероятностями $\{P_1,\ldots,P_N\}$. Вероятности захвата объектов будем считать независимыми как от их местоположения

в ИКС, так и от связи с другими объектами, и таковыми, которые являются характеристикой соответствующих объектов ИКС.

Для определения вероятностей захвата объектов проанализируем причины успешности атак. Согласно исследованиям компании TrustWave [124], основными путями доступа злоумышленников являются использование возможностей внешнего подключения к административным консолям управления, использование слабых или недостаточных настроек при соединении некоторых сегментов сетей или подключении к партнерским сетям, а также уязвимости в программном обеспечении. В табл. 5.1 приведены основные причины, которые приводят к успешным атакам внешних и внутренних злоумышленников.

Таблица 5.1

Основные уязвимости ИКС

№ п/п	Внешние уязвимости	Внутренние уязвимости
1	Unprotected Application Management Interface	Address Resolution Protocol (ARP) Cache Poisoning
2	Unprotected Infrastructure Management Interface	Microsoft SQL Server with Weak Creds for Admin
3	Access to Internal Application via the Internet	Weak Password for Admin Level System Account
4	Misconfigured Firewall Permits Access to Internal	Client Sends LM Response for NTLM Authentication
5	Default or Easy to Determine Credentials	Crypto Keys Stored Alongside Encrypted Data
6	Sensitive Information, Source Code, etc. in Web Dir	Cached Domain Credentials Enabled on Hosts
7	Static Credentials Contained in Client	NFS Export Share Unprotected
8	Domain Name Service (DNS) Cache Poisoning	Sensitive Information Transmitted Unencrypted
9	Aggressive Mode IKE Handshake Support	Sensitive Info Stored Outside Secured Zone
10	Exposed Service Version Issues (Buffer Overflows)	VNC Authentication Bypass
11	Missing Critical Non-system Software Patches	
12	Missing Critical System Software Patches	

Исходя из приведенных в таблице данных и исследований ряда специалистов в области информационной безопасности [125, 126, 127], основные причины, которые приводят к захвату объектов, можно разделить на два класса: ошибки при настройке (или недостаточные настройки) объектов и уязвимости программного обеспечения объектов.

Основываясь на этом, вероятность захвата объекта P_i можно определить как

$$P_i = P_i^s + P_i^v - P_i^s P_i^v, \tag{5.7}$$

где P_i^s – вероятность ошибки при настройке объекта или его недостаточная настройка;

P_i^v – вероятность наличия уязвимостей в программном обеспечении объекта.

Данные значения вероятностей не будут статичными и, в свою очередь, будут зависеть от ряда факторов (часть которых составляют причины, приведенные в табл. 5.1).

Вероятность P_i^s во многом определяется человеческим фактором, сложностью настроек программных или программно-аппаратных средств, а также количеством доступных для злоумышленника сервисов. С увеличением каждого из этих показателей соответственно будет возрастать вероятность нахождения злоумышленником ошибок в настройках, что приведет к возможности захвата объекта [127].

Корректность настроек проверяется во время экспертных испытаний, аудита и тестов на проникновение в ИКС. После внесения изменений в настройки соответствующего объекта значение вероятности P_i^s будет увеличиваться до тех пор, пока не будут проведены повторные испытания или аудит. Кроме этого, необходимо среди объектов ИКС выделять множество тех настроек, которые будут критичны для безопасности всей системы.

Вероятность P_i^v наличия уязвимостей в программном обеспечении зависит от качества процесса разработки и тестирования компанией-производителем продукта, а также может основываться на аналогичных статистических данных для других продуктов производства данной компании.

Статистические данные о количестве найденных уязвимостей в соответствующих продуктах доступны на сайте National Vulnerability Database (NVD) CVE Statistics.

Для примера на графике (рис. 5.4) приведено количество уязвимостей, найденных в ОС в соответствующие годы [128].

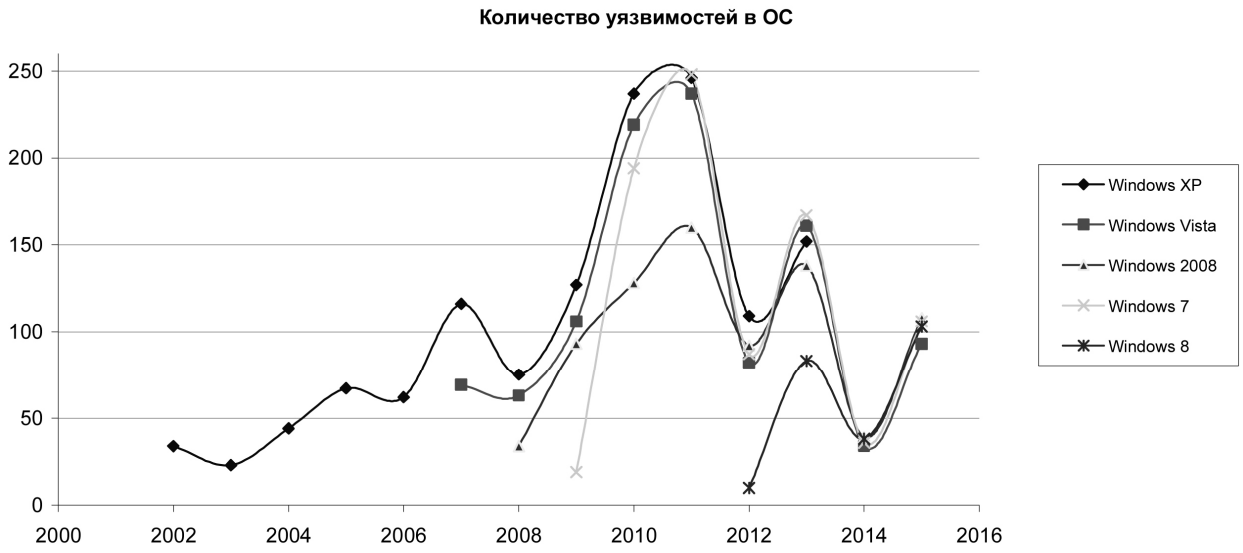


Рис. 5.4. Количество уязвимостей в ОС

Также следует отметить, что количество найденных уязвимостей в программном обеспечении со временем увеличивается, соответственно, будет увеличиваться со временем и значение P_i^v. После установления обновлений P_i^v будет уменьшаться, но потом снова будет возрастать.

Необходимо отметить, что при определении вероятности захвата источника атаки следует также учитывать вероятность его выбора злоумышленником для совершения атаки. Например, вероятность попыток совершения атак на веб-сервер из Интернет будет выше, чем изнутри сети. Таким образом, если объект принадлежит к источникам атаки ($v_i \in A$), во время вычисления вероятности его захвата на основе (5.7) будем определять, как

$$P_i = P_i^a (P_i^s + P_i^v - P_i^s P_i^v), \tag{5.8}$$

где P_i^a – вероятность попытки атаки через данный источник атаки.

Соответствующие значения вероятностей P_i^a, P_i^v, P_i^s могут определяться на основе статистических данных, путем экспертных оценок или другими методами.

При необходимости детализации или учете других факторов, которые будут влиять на значение вероятности захвата объекта, можно воспользоваться логико-вероятностным подходом. В его рамках события ($Z_i^1,...,Z_i^n$),

которые могут привести к захвату объекта v_i, рассматриваются независимо и связываются логическим символом «ИЛИ». Вероятность реализации каждого из событий обозначим как $P_i^1,...,P_i^n$. Тогда вероятность захвата объекта v_i будет определяться следующей формулой:

$$P_i = P((Z_i^1 \vee ... \vee Z_i^n) = 1) =$$
$$= \sum_{k=1}^{n} P_i^k - \sum_{k<j} P_i^k P_i^j + \sum_{k<j<m} P_i^k P_i^j P_i^m - ... + (-1)^{n-1} P_i^1 ... P_i^n . \quad (5.9)$$

5.3. Построение модели системы киберзащиты на основе комплексной модели киберугроз

На основе анализа комплексной модели киберугроз $G(V, E, A, O, P)$ среди возможных вариантов, с помощью которых можно усложнить злоумышленнику осуществление успешной кибератаки на ИКС, можно выделить следующие:

– уменьшить количество сценариев кибератак, что физически будет приводить к устранению связей между объектами сети;

– уменьшить вероятности захвата объектов, которые входят в соответствующий путь атаки;

– уменьшение количества источников или/и объектов кибератаки (поверхности атаки).

Для первого случая необходимо изменять топологию сети $G(V, E)$, а для второго – применять механизмы защиты, при помощи которых можно уменьшить вероятности захвата объектов $P = \{P_1,...,P_N\}$. Третий случай является наиболее эффективным, но требует изменений относительно функционирования ИКС и не всегда может быть приемлемым, поэтому его рассматривать не будем.

Для уменьшения вероятностей захвата объектов предлагается использовать механизм защиты M_i. При его применении вероятность захвата может быть сведена к нулю или снижена в несколько раз, что будет зависеть от характеристик самого механизма защиты. Эту характеристику механизма защиты назовем коэффициентом эффективности и обозначим как K_i. Данный коэффициент определяет вероятность того, что механизм защиты M_i сможет противостоять кибератакам, направленным на захват объекта v_i.

Под механизмом защиты может также подразумеваться некоторая деятельность, направленная на нахождение ошибок в настройках или уязвимостей в программном обеспечении: анализ настроек, анализ исходного кода, тестирование на проникновение и т. п.

С учетом механизмов защиты вероятность захвата объекта $v_i \in V$ будет определяться как

$$P_i^{'} = P_i(\overline{M_i K_i}) = P_i(1 - M_i K_i), \tag{5.10}$$

где $P_i \in [0,1]$ – вероятность захвата объекта;

$M_i = \{0,1\}$ – наличие или отсутствие механизма защиты;

$K_i \in [0,1]$ – коэффициент эффективности механизма защиты.

Коэффициенты эффективности могут определяться на основе статистических данных, путем экспертных оценок или другими методами в зависимости от типа механизма защиты (например, антивируса, межсетевого экрана, криптографического средства) и должны учитывать их свойства.

Такую модель защиты ИКС предлагается представить в виде графа защиты (рис. 5.5), который получаем путем дополнения графа кибератак механизмами защиты, которые размещаются в вершинах графа. Кортеж $G(V, E, A, O, P, M, K)$ назовем графом киберзащиты.

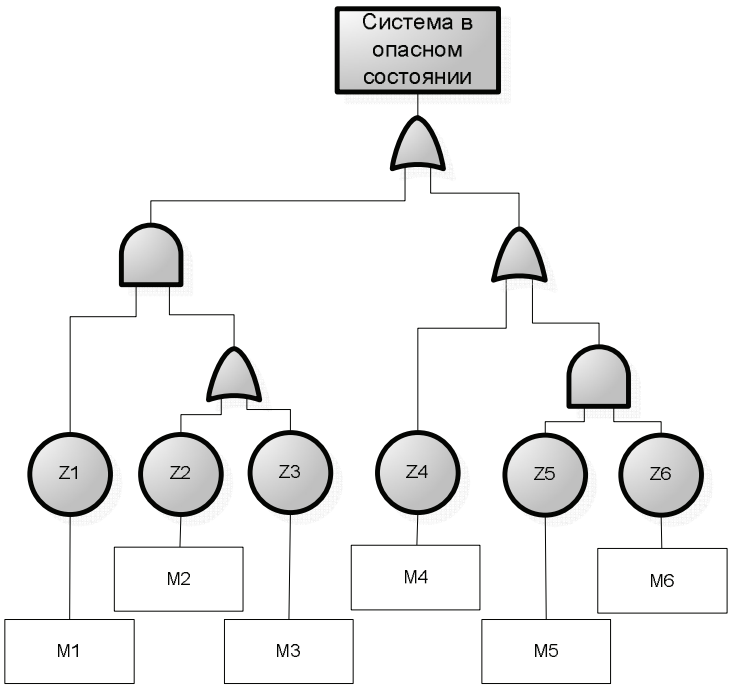


Рис. 5.5. Граф защиты

5.4. Построение функции вероятности успешности кибератаки на основе логико-вероятностного метода

Как было отмечено в первой главе, ИКС представляет собой набор сервисов, которые взаимодействуют между собой в киберпространстве. Таким образом, при разработке модели для определения уровня защищенности системы или ее составляющих объекты и сервисы системы должны рассматриваться интегрально, а также должны учитываться их связи между собой. С точки зрения архитектуры системы очевидным является вариант, что система является наиболее защищенной в случае, если каждый сервис не взаимодействует ни с одним другим, и наименее защищенной – если каждый сервис взаимодействует с каждым.

Исходя из этого, модель для оценки защищенности должна учитывать расположение в системе соответствующего сервиса и все его связи с другими сервисами.

Для анализа уровня защищенности ИКС будем использовать логико-вероятностный метод – он является достаточно универсальным и может применяться при анализе безопасности ССС, которые могут содержать циклы и повторяющиеся элементы [129, 130, 131].

Информационно-коммуникационная система является одним из видов ССС, поэтому для решения задачи оценивания уровня защищенности ИКС и защищенности ее объектов, с учетом топологии и возможных путей доступа к ней злоумышленников, будем использовать логико-вероятностный метод, предложенный академиком И. О. Рябининым [132].

Логико-вероятностные модели поведения ИКС с двух- и трехуровневым стеком протоколов были рассмотрены в гл. 3.

Фундаментальными понятиями в логико-вероятностной теории безопасности являются понятия опасного состояния системы и соответствующей логической функции, которая представляет развитие этого опасного состояния (ФОС).

В случае ИКС развитие опасного состояния будет представлять собой последовательность действий или событий, которые могут повлиять на уровень защищенности оцениваемых объектов – сценарий кибератаки. Кибератака будет считаться успешной при условии, что злоумышленнику удастся успешно выполнить все промежуточные действия сценария кибератаки.

Таким образом, для оценивания уровня защищенности объектов ИКС последовательность действий злоумышленника и все пути, которыми он может получить доступ к объектам системы, будут формировать сценарии кибератак или ФОС логико-вероятностной теории. При этом сценарий кибератаки составляет конъюнкцию событий Z_i, каждое из которых нельзя удалить, не нарушив опасного функционирования системы. Используя логико-вероятностный метод, запишем такую конъюнкцию в виде функции алгебры логики (ФАЛ):

$$\varphi_l = \bigwedge_{i \in K_{\phi l}} Z_i, \tag{5.11}$$

где K_{ϕ_l} – последовательность действий злоумышленника в информационно-коммуникационной системе, которая приводит к опасному состоянию определенного объекта системы, что соответствует *l*-му сценарию кибератаки (5.4), где:

$Z_i = 1$ – если объект v_i захвачен (подцель достигнута);

$Z_i = 0$ – если объект v_i не захвачен (подцель не достигнута).

Тогда маршрут R_l из сценария кибератаки K_φ можно представить как конъюнкцию событий в виде (5.11). Таким образом, записывается один из возможных вариантов развития сценария кибератаки.

Исходя из этого, для объекта v_i ИКС можно записать возможные кибератаки в виде ФАЛ и событий Z_j (где $j \in K_{\varphi_l}$) с помощью операций конъюнкции и дизъюнкции:

$$y_i(Z_1,...,Z_m) = \bigvee_{l=1}^{d} \varphi_l = \bigvee_{l=1}^{d} \left[\bigwedge_{i \in K_{\varphi l}} Z_i \right], \tag{5.12}$$

где $d = \left| K_\varphi \right|$ – количество сценариев кибератаки от источников атак A для объекта v_i.

Если булевая функция $y_i(Z_1,...,Z_m)$ (5.12) будет равна «1», можно говорить про успешность совершения кибератаки на объект v_i.

Согласно логико-вероятностной теории вероятность перехода объекта v_i ИКС в опасное состояние будет определяться как вероятность того, что ФАЛ (5.12) будет равна «1»:

$$P\{y_i(Z_1,...,Z_m) = 1\} = P\left\{ \bigvee_{l=1}^{d} \left[\bigwedge_{i \in K_{\varphi l}} Z_i \right] = 1 \right\}. \tag{5.13}$$

Значение функции $P\{y_i(Z_1,...,Z_m)=1\}$ будет определять вероятность реализации хотя бы одного из сценариев кибератак для объекта v_i.

Зная значения вероятностей захвата каждого из объектов (вероятности реализации каждого из событий Z_i), можно вычислить значение функции (5.13). Для этого необходимо преобразовать функцию алгебры логики (5.12) в эквивалентную ей форму: ортогональную дизъюнктивную нормальную форму, идеальную дизъюнктивную нормальную форму или бесповторную функцию в базисе конъюнкций-отрицаний [133]. Для функций, представленных в этих формах, можно выполнить прямое замещение булевых переменных Z_i на их вероятностные значения $P\{Z_i=1\}=P_i$ – вероятности захвата объектов:

$$P\{y_i(Z_1,...,Z_N)=1\}=P(P_1,...,P_N). \qquad (5.14)$$

Таким образом, для каждого из объектов ИКС может быть построена функция вероятности (5.14), которая будет определять уровень защищенности объекта v_i ИКС и зависеть от вероятностей захвата промежуточных объектов, которые входят в сценарий кибератаки.

Довольно часто возникает задача определить не только уровень защищенности отдельных объектов ИКС, а также оценить интегральный уровень защищенности ИКС в целом.

Уровень защищенности ИКС будем определять, исходя из защищенности ее ключевых объектов, которые принадлежат множеству объектов кибератак $O=(o_1,...,o_M)\subset V$. Будем считать, что система перейдет в опасное состояние, если в опасное состояние перейдет хотя бы один из ее ключевых объектов $O=(o_1,...,o_M)$.

Исходя из логико-вероятностной теории, запишем ФОС ИКС в виде

$$\begin{gathered} y(Z_1,...,Z_N)=\bigvee_{i=1}^{M} y_i(Z_1,...,Z_N); \\ y_i(Z_1,...,Z_N)=\bigvee_{l=1}^{d_i}\left[\bigwedge_{j\in K^i{}_{\phi l}} Z_j\right], \end{gathered} \qquad (5.15)$$

где $y_i(Z_1,...,Z_N)$ – ФОС o_i-го объекта системы;

M – количество ключевых объектов системы.

Булевая функция $y(Z_1,...,Z_N)$ строится на основе графа кибератак и, согласно логико-вероятностному методу, называется функцией опасного состояния системы ИКС. Значение данной функции будет равно «1» ($y(Z_1,...,Z_N) = 1$) в случае успешности кибератаки хотя бы на один из важных объектов ИКС. Но основе функции опасного состояния (5.15) и (5.13) уровень защищенности ИКС определяется как вероятность реализации хотя бы одного из сценариев кибератак, т. е.

$$P\{y(Z_1,...,Z_N)=1\} = P\left\{\bigvee_{i=1}^{M} y_i(Z_1,...,Z_N)=1\right\}. \quad (5.16)$$

Для вычисления значения функции (5.16) необходимо функцию алгебры логики (5.15) преобразовать в эквивалентную ей ортогональную дизъюнктивную нормальную форму и для функции, представленной в такой форме, выполнить прямое замещение булевых переменных Z_i на их вероятностные значения $P\{Z_i=1\} = P_i$:

$$P\{y(Z_1,...,Z_N)=1\} = P(P_1,...,P_N). \quad (5.17)$$

Функцию вероятности (5.17) назовем **функцией вероятности успешности кибератаки.** Основным отличием от функции оценки уровня кибернетической безопасности, представленной в гл. 3, является то, что (5.17) учитывает топологию сети при моделировании киберугроз на ИКС.

Общий алгоритм для оценки уровня защищенности с использованием логико-вероятностного метода состоит из пяти этапов и приведен на рис. 5.6 [134]:

1. Разработка вербальной модели – описание условий безопасного функционирования системы.

2. Разработка графической модели – формализованное наглядное описание системы в виде графа.

3. Преобразование графической модели в логическую – построение функции алгебры логики.

4. Вероятностная модель – конечное выражение для совершения вычислений.

5. Выполнение расчетов.

После того, как произведены расчеты и проведен анализ результатов, может появиться необходимость уточнения моделей и входных значений.

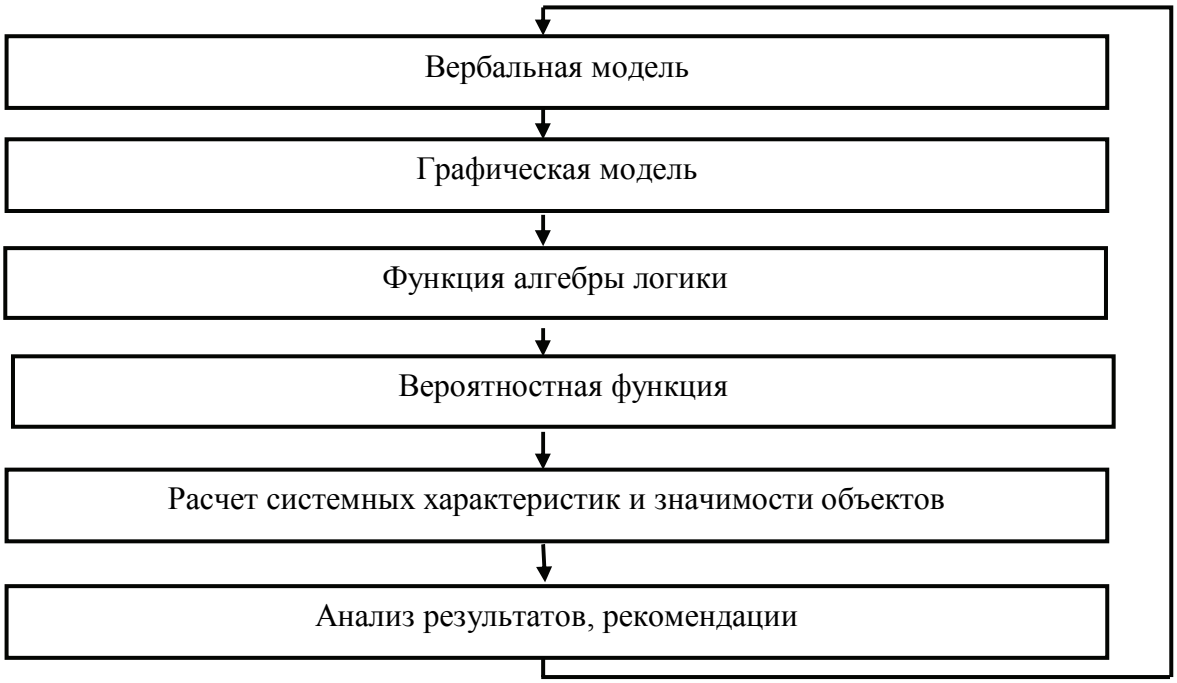


Рис. 5.6. Алгоритм оценки уровня защищенности ИКС

5.5. Приведение функции вероятности успешности кибератаки в виде ФАЛ к ортогональной дизъюнктивной нормальной форме

Используя логико-вероятностный метод для оценки уровня защищенности ИКС или ее объектов, необходимо осуществлять переход от ФАЛ опасного состояния к ее вероятностному виду. Такой переход можно совершить прямой заменой булевых переменных на их вероятностные значения. Но для этого ФАЛ должна быть записана в одной из форм: ортогональной дизъюнктивной нормальной форме, идеальной дизъюнктивной нормальной форме или бесповторной функции в базисе конъюнкций-отрицаний [133].

Для автоматизации построения ФОС и перехода к функции вероятности защищенности разработаем алгоритм для перехода к ортогональной дизъюнктивной нормальной форме (ОДНФ).

Будем считать, что на вход алгоритма подаются только ФАЛ, в которых отрицание относится только к отдельным переменным.

Работа алгоритма по преобразованию в ОДНФ состоит из следующих этапов:

А. Построение обратной польской нотации для формулы.

Б. Построение дерева формулы.

В. Преобразование и обход дерева для получения формулы в дизъюнктивной нормальной форме (ДНФ).

Г. Ортогонализация полученной ДНФ.

Рассмотрим каждый из этапов подробнее.

Этап А.

Для получения обратной польской нотации формулы используем алгоритм Дейкстри для двух операций: конъюнкции ($\wedge$) и дизъюнкции ($\vee$). Символ отрицания (') рассмотрим как часть имени переменной (операнд).

Вводим приоритет операций:

(	0
$\vee$	1
$\wedge$	2

Работа алгоритма:

Пока не достигнут конец входной строки, читать очередную лексему и выполнить над ней следующие операции:

1. Если прочитан операнд, записываем его во входную строку.

2. Если прочитана открывающаяся скобка – «(» (левая скобка), поместить ее в стек.

3. Если прочитана закрывающаяся скобка – «)» (правая скобка), перенести из стека во входную строку все до левой скобки, при этом сами скобки уничтожаются (удаляются из стека и не переносятся во входную строку).

4. Если прочитан знак операции, перенести из стека во входную строку все операции с большим или таким же приоритетом, а прочитанную операцию поместить в стек.

Если достигнут конец входной строки, перенести все лексемы из стека во входную строку и завершить работу.

Этап Б.

Используя польскую запись, строим бинарное дерево, которое отвечает данной формуле. Например, для формулы $(A \vee B) \wedge (C \vee D)$, польская запись которой имеет вид $AB \vee CD \vee \wedge$, получаем дерево, показанное на рис. 5.7.

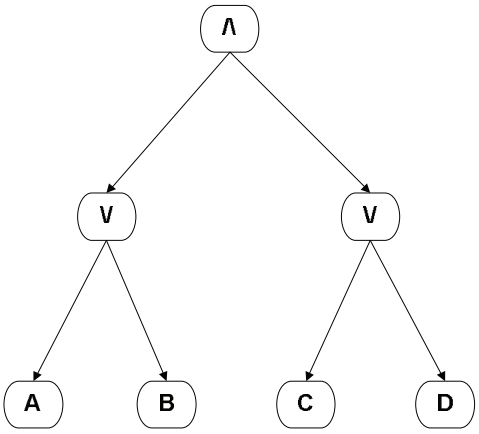


Рис. 5.7. Дерево, которое соответствует формуле $(A \vee B) \wedge (C \vee D)$

Работа алгоритма:

Рассмотрим польскую запись формулы справа налево. Пока не прочитана вся строка:

Шаг 1. Читаем очередную лексему.

Шаг 2. Создаем вершину дерева с таким именем.

Шаг 3. Если ранее не было создано ни одной вершины, делаем текущую вершину корнем дерева и переходим на Шаг 1.

Шаг 4. Если предыдущая созданная вершина соответствует операции ($\wedge$ или $\vee$) и не имеет правого сына, то делаем текущую вершину ее правым сыном; переходим на Шаг 1.

Шаг 5. Если предыдущая созданная вершина соответствует операции ($\wedge$ или $\vee$) и имеет правого сына, то делаем текущую вершину ее левым сыном, переходим на Шаг 1.

Шаг 6. Если предыдущая созданная вершина соответствует переменной, то находим среди созданных последнюю вершину, которая соответствовала операции и не имела левого сына, и делаем текущую вершину ее левым сыном, переходим на Шаг 1.

Если возникла ситуация, когда работа алгоритма еще не завершена, а переход ни по одному из правил невозможен, это свидетельствует о том, что формула является синтаксически неправильной.

Этап В. Преобразование дерева.

Выполняем поворот по следующим правилам:

Шаг 1. Находим вершину типа «$\wedge$» (обозначим ее О), которая имеет правого (левого) сына типа «$\vee$» (обозначим поддерево, которое соответствует

этому сыну – S1, а поддерево, которое соответствует другому сыну O – S2). Если такая вершина не найдена, завершаем работу.

Шаг 2. Создаем новую вершину типа «∨» – обозначим ее P.

Шаг 3. Добавляем P двух сыновей типа «∧» – обозначим их R1 и L1.

Шаг 4. Левым (правым) сыном R1 делаем S2.

Шаг 5. Левым (правым) сыном L1 делаем S2.

Шаг 6. Правым (левым) сыном R1 делаем правого (левого) сына S1.

Шаг 7. Правым (левым) сыном L1 делаем левого (правого) сына S1.

Шаг 8. Заменяем O созданной вершиной. Переходим на Шаг 1.

Для формулы, которая рассматривалась ранее, понадобится две итерации. Вид дерева после каждой из них показан на рис. 5.8.

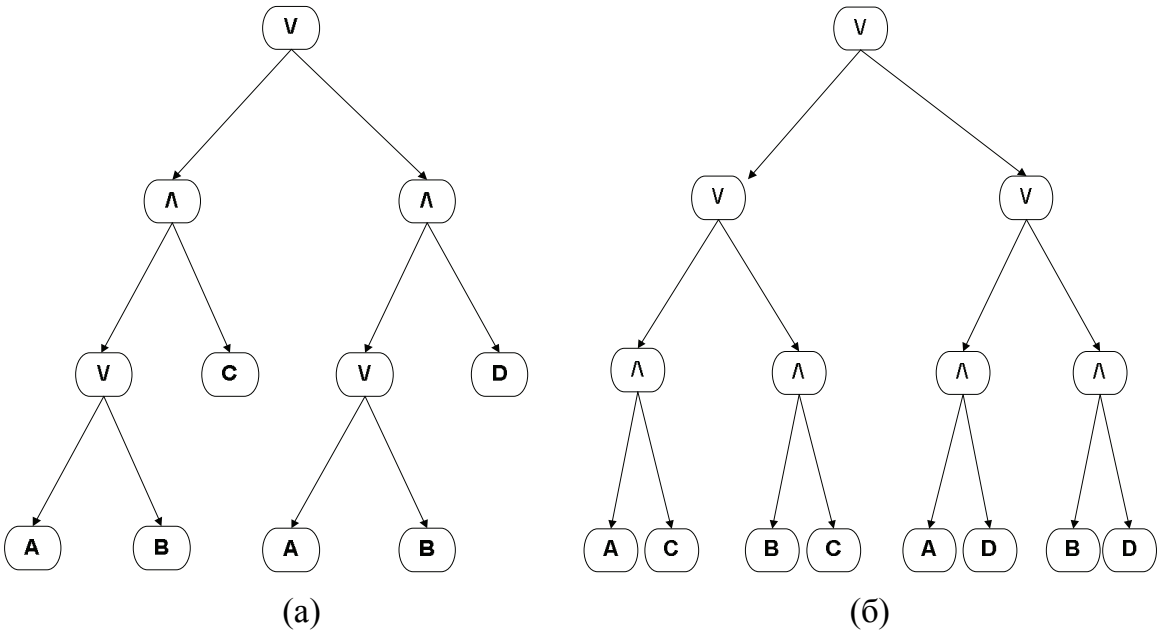


Рис. 5.8. Результат поворота дерева: *а* – первая итерация; *б* – вторая итерация

Таким образом, мы, по сути, раскрыли скобки в формуле. После поворота дерева можем выполнить его обход так, чтобы получить набор элементарных конъюнкций, которые входят в ДНФ начальной формулы.

Обход дерева:

Шаг 0. Отмечаем все вершины как непосещенные.

Шаг 1. Текущая вершина (P) – корень дерева. Если все листья дерева посещены, завершаем работу, иначе начинаем запись новой конъюнкции.

Шаг 2. Если P имеет правого сына, делаем его текущей вершиной и переходим на Шаг 3.

Шаг 3. Если Р соответствует переменной, добавляем ее имя в конъюнкцию, отмечаем Р як посещенную, переходим к Шагу 7.

Шаг 4. Если Р соответствует дизъюнкции и ее правый сын равен «0» или Р соответствует конъюнкции, то делаем текущей вершиной левого сына Р и обходим поддерево, корнем которого он является.

Шаг 5. Если Р – корень дерева и у этой вершины есть только правый сын, удаляем правого сына Р (тут и далее имеется ввиду удаление всего поддерева, корнем которого является данная вершина).

Шаг 6. Если Р – корень дерева и все листья его сыновей посещены – удаляем сыновей Р.

Шаг 7. Если Р соответствует дизъюнкции и имеет правого сына и все листья правого сына уже посещены, удаляем правого сына Р.

Шаг 8. Если Р соответствует дизъюнкции и имеет левого сына и все листья левого сына Р уже посещены, удаляем левого сына Р.

В результате работы алгоритма получаем набор элементарных конъюнкций, которые и составляют ДНФ начальной формулы.

Этап Г.

Используем алгоритм ортогонализации логической формулы, описанный и обоснованный в [132].

На вход алгоритма подается массив конъюнкций, которые входят в ДНФ начальной формулы $\begin{vmatrix} K_1 \\ \dots \\ K_n \end{vmatrix}$.

Шаг 1. Выполняем сортировку элементарных конъюнкций в порядке увеличения количества членов конъюнкции.

Шаг 2. Для каждой элементарной конъюнкции $K = x_1 \wedge \dots \wedge x_n$, кроме последней (самой длинной), формируем отрицание в виде $K' = \begin{vmatrix} x_1' \\ x_1 \wedge x_2' \\ \dots \\ x_1 \wedge \dots \wedge x_{n-1} x_n' \end{vmatrix}$.

Шаг 3. Упрощаем полученные отрицания: удаляем противоречивые конъюнкции.

Шаг 4. Формируем ОДНФ в виде $\begin{vmatrix} K_1' \\ K_1 K_2' \\ \dots \\ K_1 \dots K_{n-1} K_n' \end{vmatrix}$.

Шаг 5. Удаляем повторы переменных в конъюнкциях.

Шаг 6. Удаляем повторы конъюнкций.

Шаг 7. Удаляем противоречивые конъюнкции.

5.6. Исследование структуры функции вероятности успешности кибератаки

Исходя из того, что функция вероятности успешности кибератаки (5.17) определяется как вероятность того, что функция алгебры логики (5.15), построенная на основе графа кибератак (5.6), будет равна единице, следует, что функция вероятности успешности кибератаки не будет иметь фиксированный вид. Структура данной функции будет зависеть от топологии сети ИКС, а также от выбранных источников и объектов кибератак, т. е. при одинаковой топологии сети $G(V,E)$, но при разных графах кибератак $G(V,E,A,O)$ (разные объекты или источники атак) структура функции может быть разной.

В общем случае полученная вероятностная функция (5.17) – функция вероятности успешности кибератаки – зависит от переменных $\{P_1,...,P_N\}$, а ее структура зависит от параметров G,A,O:

$$P(G,A,O,P_1,...,P_N), \tag{5.18}$$

где $G(V,E)$ – топология сети ИКС;

$A \subset V$ – множество источников кибератак;

$O \subset V$ – множество объектов кибератак;

$P_1,...,P_N$ – вероятности захвата объектов.

Таким образом, изменяя параметры данной функции, можно уменьшать или увеличивать значения вероятности успешности кибератаки на систему.

Исходя из этого, используя функцию вероятности успешности кибератаки (5.18) в качестве критерия уровня защищенности ИКС, можно выполнить построение безопасной топологии сети *G*, а также оптимальным образом разместить механизмы защиты для уменьшения вероятностей захвата объектов

$P_1,...,P_N$ по формуле (5.10). Этому будут посвящены следующие пункты данной главы.

Исходя из изменчивости функции вероятности успешности кибератаки, довольно тяжело исследовать ее свойства в общем случае, так как ее структура будет непосредственно зависеть от источников и объектов кибератак, а также от топологии сети информационно-коммуникационной системы.

По своей структуре функция вероятности успешности кибератаки $P(P_1,...,P_N)$ представляет собой полилинейную форму [135] следующего вида:

$$F(x_1,x_2,...,x_n)=\sum_{\nu_1,\nu_2,...,\nu_n\in\{0,1\}} C_{\nu_1,\nu_2,...,\nu_n}x_1^{\nu_1}x_2^{\nu_2}...x_n^{\nu_n},$$

или в терминах вероятностей захвата объектов ИКС

$$P(P_1,P_2,...,P_n)=\sum_{\nu_1,\nu_2,...,\nu_n\in\{0,1\}} C_{\nu_1,\nu_2,...,\nu_n}P_1^{\nu_1}P_2^{\nu_2}...P_n^{\nu_n}=$$

$$=\sum_{\nu_1,\nu_2,...,\nu_n\in\{0,1\}} A_{\nu_1,\nu_2,...,\nu_n}P_1^{\nu_1}P_2^{\nu_2}...P_n^{\nu_n}+\sum_{\nu_1,\nu_2,...,\nu_n\in\{0,1\}} B_{\nu_1,\nu_2,...,\nu_n}(1-P_1)^{\nu_1}(1-P_2)^{\nu_2}...(1-P_n)^{\nu_n},$$

где A, B, C – некоторые коэффициенты.

То есть, данная функция $P(P_1,...,P_N)$ будет линейна по каждому из своих аргументов при фиксированных значениях всех остальных. При фиксации всех аргументов, кроме P_k, функция вероятности успешности кибератаки имеет вид

$$P(P_1,P_2,...,P_n)=A_kP_k+B_k(1-P_k)+C_k=D_kP_k+E_k, \qquad (5.19)$$

при этом

$$\forall k: A_kP_k+B_k(1-P_k)+C_k\le 1.$$

Коэффициенты $C_{\nu_1,\nu_2,...,\nu_n}$ (соответственно $A_{\nu_1,\nu_2,...,\nu_n}$ и $B_{\nu_1,\nu_2,...,\nu_n}$) будут меняться в зависимости от ИКС и, как уже отмечалось, зависеть от источников A, объектов кибератаки O и топологии сети $G(V,E)$.

Также следует учесть свойство логико-вероятностного метода, которое говорит о том, что если один маршрут сценария атаки является частью другого, то из них остается только тот, который короче, т. е., если имеем маршруты на графе S_1 и S_2, и $S_1\subset S_2$, то остается только маршрут S_1. Это следует из того, что данный маршрут является самым коротким сценарием осуществления атаки.

Покажем это на примере.

Если

$$S_1 = \{a, b, c, d, e\} \text{ и } S_2 = \{b, c, d\},$$

то

$$y(Z_a, Z_b, Z_c, Z_d, Z_e) = y_1(Z_a, Z_b, Z_c, Z_d, Z_e) \vee y_2(Z_b, Z_c, Z_d) = \\ = Z_a Z_b Z_c Z_d Z_e \vee Z_b Z_c Z_d = Z_b Z_c Z_d (1 \vee Z_a Z_e) = Z_b Z_c Z_d .$$

Таким образом, есть смысл рассматривать только те маршруты сценариев кибератак, которые не совпадают.

5.7. Доказательство свойств монотонности функции вероятности успешности кибератаки

Для решения задач построения безопасной топологии и оптимального размещения механизмов защиты докажем, что функция вероятности успешности кибератаки не возрастает при удалении одного или нескольких ребер из графа кибератак, а также монотонность по каждой из переменных.

Дуго-монотонность функции вероятности успешности кибератаки. Сначала дадим общее определение дуго-монотонности.

Определение согласно [136, 137]: функционал F называется дуго-монотонным, если для произвольных графов $G_1 =< V, E_1 > \in \Omega$, $G_2 =< V, E_2 > \in \Omega$, таких что $E_1 \subset E_2$, выполняется неравенство $F(G_1) \le F(G_2)$ (т. е. при удалении из графа дуг значение функционала не увеличивается).

Покажем, что функция вероятности успешности кибератаки будет дуго-монотонной, т. е. не будет возрастать при удалении одного или нескольких ребер из графа атак.

Это утверждение в наших терминах будет формулироваться следующим образом: функция $P(G)$ будет дуго-монотонной, если для произвольных графов атак $G_1 =< V, E_1, A, O > \in \Omega$, $G_1 =< V, E_2, A, O > \in \Omega$, где $O \subset V$ – объекты, а $A \subset V$ – источники атак, таких что $E_1 \subset E_2$, выполняется неравенство $P(G_1) \le P(G_2)$.

При фиксированных источниках и объектах атаки функция вероятности будет зависеть только от топологии, и вместо $E_1 \subset E_2$ будем записывать $G_1 \subset G_2$.

Неформальным доказательством свойства дуго-монотонности $P(G)$ могут быть следующие соображения.

При создании новой связи между объектами сети будет увеличиваться количество возможных сценариев кибератак, а также, соответственно, путей атак на графе кибератак и, таким образом, будет увеличиваться значение функции вероятности успешности кибератаки. При уничтожении связи количество путей атаки будет уменьшаться и значение функции вероятности успешности кибератаки будет также уменьшаться.

Теперь, в следующем утверждении, приведем формальное доказательство дуго-монотонности функции вероятности успешности кибератаки.

Утверждение 1. Функция вероятности успешности кибератаки является дуго-монотонной и не возрастает при удалении одного или нескольких ребер из графа атак.

Доказательство. Пусть $X = \{x_1, ..., x_t\}$ – множество логических переменных, которые соответствуют захвату отдельных объектов системы. Запишем ФАЛ, которая соответствует реализации опасного состояния для системы. Для этого пересчитаем все возможные ацикличные пути атак. Функция алгебры логики имеет вид

$$y_1 = \bigcup_{i=1}^{m} \bigcap_{j=1}^{n_i} x_{ij}, де \; \mathrm{x_{ij}} \in X, \forall i, j \, . \tag{5.20}$$

Рассмотрим, что произойдет с ИКС после удаления из нее одной из связей между ее объектами, через которую проходит один из путей кибератаки (т. е. удаление одного ребра из графа кибератак). Это приведет к тому, что из (5.20) удалится одна или несколько конъюнкций, которые были в ней благодаря связи между объектами.

Докажем, что при удалении из (5.20) одной конъюнкции вероятность опасного состояния для системы не возрастает.

Без ограничения общности можно считать, что отбрасываем последнюю конъюнкцию. Получаем выражение

$$y_2 = \bigcup_{i=1}^{m-1} \bigcap_{j=1}^{n_i} x_{ij} \, . \tag{5.21}$$

Теперь ФОС (5.20) и (5.21) записаны в дизъюнктивной нормальной форме. Выполним ортогонализацию и приведем их к ОДНФ.

Введем обозначение для элементарной конъюнкции $K_i = \bigcap_{j=1}^{n_i} x_{ij}$. Тогда ортогональная форма для (5.20) будет иметь вид

$$y_1^* = \bigcup_{i=1}^{m} (K_i \cap \bigcap_{j=1}^{i-1} K_j^{'}). \tag{5.22}$$

Ортогональная форма для (5.21):

$$y_2^* = \bigcup_{i=1}^{m-1} (K_i \cap \bigcap_{j=1}^{i-1} K_j^{'}). \tag{5.23}$$

Как видим, (5.22) и (5.23) связаны соотношением

$$y_1^* = y_2^* + K_m \cdot \bigcap_{j=1}^{m-1} K_j^{'}. \tag{5.24}$$

Ортогональная дизъюнктивная нормальная функция является формой перехода к полному замещению, значит соотношение (5.24) будет сохраняться для вероятностей:

$$P(y_1) = P(y_2) + P(K_m \cdot \bigcap_{j=1}^{m-1} K_j^{'}).$$

В силу неотрицательности вероятности будем иметь:

$$P(y_1) \geq P(y_2).$$

Таким образом, было доказано, что при удалении одной конъюнкции из ФОС, записанной в виде ФАЛ вида (5.20), вероятность истинности этой ФОС не увеличивается. Последовательно удаляя конъюнкции из ФОС (при этом вероятность истинности каждый раз не возрастает), доказываем, что при удалении ребра из графа сети ИКС значение функции вероятности успешности кибератаки возрастать не будет.

Таким образом, доказано дуго-монотонность функции вероятности успешности кибератаки:

$$\forall G_1, G_2 \in \Omega,\ G_1 \subset G_2 \ \Rightarrow\ P(G_1) \leq P(G_2).$$

Отметим, что во время доказательства этого утверждения мы не использовали особенности модели ИКС, поэтому данное утверждение может быть справедливым и в более общем случае.

Вершино-монотонность функции вероятности успешности кибератаки. Исходя из дуго-монотонности функции вероятности, можно увидеть, что она также будет и вершино-монотонна, т. е. при удалении как изолированного, так и связанного с другими объекта, функция вероятности возрастать не будет. Это следует из того, что если объект является изолированным, то функция вероятности вообще не будет зависеть от вероятности его захвата, а если он имеет несколько связей с другими объектами сети, то, последовательно исключая их и пользуясь свойством дуго-монотонности, сведем его к изолированному.

Монотонность функции вероятности успешности кибератаки по каждой из переменных. Рассмотрим следующее утверждение.

Утверждение 2. Функция вероятности успешности кибератаки при фиксированной топологии сети является монотонно возрастающей по каждой своей переменной:

$$\forall P_i^{(1)}, P_i^{(2)},\ P_i^{(1)} < P_i^{(2)} \ \Rightarrow\ P(P_1,...,P_i^{(1)},...,P_N) \le P(P_1,...,P_i^{(2)},...,P_N).$$

Доказательство. Сначала покажем, что данное утверждение будет выполняться при $P_i^{(1)} = 0.$ Для этого вспомним определение вероятности захвата объекта в терминах алгебры логики: $P_i = P\{Z_i = 1\}$, т. е., если всегда будет $P_i = P\{Z_i = 1\} = 0$, то, соответственно, будет справедливо $P_i = P\{Z_i = 0\} = 1$ и отсюда $Z_i = 0$.

Если $Z_i = 0$, то это будет означать, что все связи, которые идут к объекту v_i, могут быть исключены из графа сети и, соответственно, из графа кибератак. Из этого будет следовать, что $G\big|_{P_i=0} \subset G\big|_{P_i\neq 0}$ и, согласно утверждению 1, $P(G)\big|_{P_i=0} \le P(G)\big|_{P_i\neq 0}$.

Теперь вспомним, что функция вероятности успешности кибератаки имеет полилинейную форму (5.19), т. е. является линейной по каждой из своих аргументов и может быть записана в виде $AP_i + C$.

Посчитаем следующую разность:

$$\begin{aligned} P(P_1,...,P_i^{(2)},...,P_N) - P(P_1,...,P_i^{(1)},...,P_N) &= P(G)\Big|_{P_i^{(2)}} - P(G)\Big|_{P_i^{(1)}} = \\ &= AP_i^{(2)} + C - AP_i^{(1)} - C = A(P_i^{(2)} - P_i^{(1)}). \end{aligned} \qquad (5.25)$$

Воспользовавшись тем, что $P(G)\big|_{P_i=0} \le P(G)\big|_{P_i\neq 0}$, и подставив в соотношение (5.25) $P_i^{(1)} = 0,$ получим, что $AP_i^{(2)} \ge 0$ и, соответственно, $A \ge 0.$

Из этого следует, что при $P_i^{(1)} < P_i^{(2)}$ всегда будет выполняться $P(G)\Big|_{P_i^{(2)}} - P(G)\Big|_{P_i^{(1)}} = A(P_i^{(2)} - P_i^{(1)}) \ge 0,$ что и надо было доказать.

Таким образом, были доказаны свойства функции успешности кибератаки, связанные с монотонностью, которые будут использованы во время решения задач построения безопасной топологии и оптимального размещения механизмов защиты.

5.8. Определение структурной значимости объектов ИКС

Изменение уровня защищенности разных объектов ИКС будет иметь разное влияние на уровень защищенности системы в целом. Во время построения ИКС для принятия и обоснования решений касательно обеспечения защищенности ее объектов необходимо иметь объективную оценку того, какой вклад вносит каждый из ее объектов в общий уровень защищенности ИКС.

Предложим подход, при помощи которого можно на основе функции вероятности успешности кибератаки оценить значимость влияния вероятностей захвата объектов на уровень защищенности ИКС.

Математическая постановка данной задачи будет состоять в нахождении вектор-функции чувствительности:

$$\xi = (\xi_1, ..., \xi_N), \tag{5.26}$$

где функции чувствительности ξ_k (или коэффициенты влияния параметров) определяются как частная производная по соответствующим параметрам $P_1, ..., P_N$ функции вероятности успешности кибератаки [138]:

$$\xi_k = \frac{\partial P(P_1, ..., P_N)}{\partial P_k}, \tag{5.27}$$

где $P(P_1, ..., P_N)$ – функция вероятности успешности кибератаки;

N – количество объектов ИКС;

P_i – вероятность захвата *i*-го объекта системы.

Исходя из полилинейной формы функции вероятности успешности кибератаки (5.19), функция чувствительности будет определяться исключительно

коэффициентами, которые стоят перед соответствующим параметром, по которому берется производная:

$$P(P_1, P_2, ..., P_N) = \sum_{\nu_1,\nu_2,...,\nu_n \in \{0,1\}} A_{\nu_1,\nu_2,...,\nu_n} P_1^{\nu_1} P_2^{\nu_2} ... P_n^{\nu_n} +$$

$$+ \sum_{\nu_1,\nu_2,...,\nu_n \in \{0,1\}} B_{\nu_1,\nu_2,...,\nu_n} (1-P_1)^{\nu_1} (1-P_2)^{\nu_2} ... (1-P_n)^{\nu_n};$$

$$\xi_k = \frac{\partial P(P_1,...,P_N)}{\partial P_k} = \tag{5.28}$$

$$= \frac{\sum_{\nu_1,\nu_2,...,\nu_n \in \{0,1\}} A_{\nu_1,\nu_2,...,\nu_n} P_1^{\nu_1} P_2^{\nu_2} ... P_n^{\nu_n} + \sum_{\nu_1,\nu_2,...,\nu_n \in \{0,1\}} B_{\nu_1,\nu_2,...,\nu_n} (1-P_1)^{\nu_1} (1-P_2)^{\nu_2} ... (1-P_n)^{\nu_n}}{\partial P_k} =$$

$$= \frac{A_k P_k + B_k (1-P_k) + C}{\partial P_k} = A_k - B_k,$$

где

$$A_k = \sum_{\nu_1,\nu_2,...,\nu_n \in \{0,1\}} A'_{\nu_1,\nu_2,...,\nu_n} P_1^{\nu_1} P_2^{\nu_2} ... P_{k-1}^{\nu_{k-1}} P_{k+1}^{\nu_{k+1}} ... P_n^{\nu_n};$$

$$B_k = \sum_{\nu_1,\nu_2,...,\nu_n \in \{0,1\}} B'_{\nu_1,\nu_2,...,\nu_n} (1-P_1)^{\nu_1} (1-P_2)^{\nu_2} ... (1-P_{k-1})^{\nu_{k-1}} (1-P_{k+1})^{\nu_{k+1}} ... (1-P_n)^{\nu_n}.$$

Для примера рассмотрим следующую функцию вероятности успешности кибератаки, записав ее в идее (5.19) для вероятностей P_3:

$$P(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = [P_4(1-P_5)(1-P_6)(1-P_7) +$$

$$+P_4P_5(1-P_6)(1-P_7) + P_1(1-P_4)(1-P_5) + P_1(1-P_4)P_5(1-P_6)]P_3 +$$

$$+[P_1P_2(1-P_4)(1-P_5)P_7 + P_1P_2(1-P_4)P_5(1-P_6)P_7](1-P_3) +$$

$$+P_4P_7 + (1-P_4)P_5P_6 + P_4P_5P_6(1-P_7) + P_4(1-P_5)P_6(1-P_7) =$$

$$= A_3P_3 + B_3(1-P_3) + C_3.$$

Тогда функция чувствительности по параметру P_3 будет иметь следующий вид:

$$\xi_3 = \frac{\partial P^S(P_1,...,P_7)}{\partial P_3} = A_3 - B_3 =$$

$$= [P_4(1-P_5)(1-P_6)(1-P_7) + P_4P_5(1-P_6)(1-P_7) +$$

$$+P_1(1-P_4)(1-P_5) + P_1(1-P_4)P_5(1-P_6)] -$$

$$-[P_1P_2(1-P_4)(1-P_5)P_7 + P_1P_2(1-P_4)P_5(1-P_6)P_7].$$

Таким образом, если известны значения вероятностей захвата объектов ИКС, то достаточно легко можно определить значимость этих объектов. Но во время проектирования ИКС или их анализа значения вероятностей захвата P_i не всегда известны.

Для этого случая предложим подход, который позволит определять значимость объектов только по структуре (виду) функции вероятности успешности кибератаки – так называемую ***структурную значимость объекта***.

Для этого воспользуемся теоремой, которая была доказана в [132], о том, что частная производная от вероятностей истинности монотонной функции алгебры логики $y(Z_1,...,Z_N)=1$ по вероятности истинности аргумента Z_k численно равна вероятности истинности булевой разности этой функции по аргументам Z_k:

$$\xi_k = \frac{\partial P\left(y(Z_1,...,Z_N)=1\right)}{\partial P(Z_k=1)} = P\left\{\Delta_{Z_k} y(Z_N)=1\right\}, \tag{5.29}$$

в которой $\Delta_{Z_k} y(Z_N)$ – булевая разность двух функций алгебры логики,

$$\Delta_{Z_k} y(Z_1,...,Z_N) = y_1^{(k)}(Z_1,...,Z_N) \oplus y_0^{(k)}(Z_1,...,Z_N),$$

где $y_1^{(k)}(Z_1,...,Z_N) = y(Z_1,...,1,...,Z_N)$ – единичная функция по отношению к аргументу Z_k,

$y_0^{(k)}(Z_1,...,Z_N) = y(Z_1,...,0,...,Z_N)$ – нулевая функция по отношению к аргументу Z_k.

Также в [110] показано, что вес объекта, который определяется по формуле

$$g_{Z_k} = P\left\{\Delta_{Z_k} y(Z_N)=1\right\}\Big|_{P_1=...=P_N=0,5}, \tag{5.30}$$

будет полностью количественно совпадать со структурной значимостью объекта:

$$B(k) = \frac{\partial P(P_1,...,P_N)}{\partial P_k}\Big|_{P_1=P_2=...=P_N=0,5}. \tag{5.31}$$

Таким образом, для вычисления структурной значимости объекта ИКС можно сначала вычислить аналитическое значение частичной производной от функции вероятности успешности кибератаки (функцию чувствительности), как это было показано в формуле (5.28), а потом вместо всех вероятностей захвата объектов подставить значение 0,5:

$$\xi_k = \frac{\partial P(P_1,...,P_N)}{\partial P_k}\Big|_{P_1=P_2=...=P_N=0,5}.$$

Но для большой системы нахождение аналитического вида частичной производной может быть достаточно громоздким, а если вычисление выполняется с помощью компьютера, то и совсем ненужным.

Для того чтобы обойтись без нахождения аналитического вида частичной производной, воспользуемся свойством о том, что количественно структурная значимость объекта (5.31) совпадает с весом объекта (5.30):

$$\xi_k = \frac{\partial P(P_1,...,P_N)}{\partial P_k}\Big|_{P_1=...=P_N=0,5} = P\{\Delta_{Z_k} y(Z_N)=1\}\big|_{P_1=...=P_N=0,5}. \quad (5.32)$$

Далее вспомним, что $P\{Z_i=1\}=P_i$, т. е. при $Z_k=1$ будем иметь $P_k=P\{1=1\}=1$, а при $Z_k=0$ будем иметь $P_k=P\{0=1\}=0$.

Как отмечено в [139], из этого следует, что

$$P\{\Delta_{Z_k} y(Z_N)=1\} = P\{(y(Z_1,...,Z_{k-1},1,Z_{k+1},...,Z_N)\oplus y(Z_1,...,Z_{k-1},0,Z_{k+1},...,Z_N))=1\} = \\ = P(P_1,...,P_N)\big|_{P_k=1} - P(P_1,...,P_N)\big|_{P_k=0}. \quad (5.33)$$

Тогда, воспользовавшись выражениями (5.32) и (5.33), получим следующее общее выражение для вычисления структурной значимости объекта ИКС по функции вероятности успешности кибератаки:

$$\xi_k = P(P_1,...,P_k,...,P_N)\big|_{P_1=...=P_{k-1}=P_{k+1}=...=P_N=0,5;P_k=1} - P(P_1,...,P_k,...,P_N)\big|_{P_1=...=P_{k-1}=P_{k+1}=...=P_N=0,5;P_k=0}. \quad (5.34)$$

5.9. Процедура построения функции вероятности успешности кибератаки

Приведем процедуру для построения функции вероятности успешности кибератаки и определения уровня защищенности ИКС (в следующем пункте продемонстрируем ее применение на примере):

1. Определить множество объектов/ресурсов/сервисов ИКС $V=\{v_1,...,v_N\}$.

2. Представить логическую топологию сети ИКС в виде графа $G(V,E)$.

3. Определить:

а) категории злоумышленников и множество источников кибератак $A=(a_1,...,a_K)\subset V$;

б) множество объектов кибератак $O=(o_1,...,o_M)\subset V$.

(*При определении множества О неявно проводится анализ рисков и выбираются те объекты, убытки от успешной кибератаки на которые будут*

наибольшими. При этом также предполагается приблизительное равенство указанных убытков в случае успешной кибератаки).

4. Определить вероятности захвата объектов, которые входят в сценарии кибератак, и построить граф кибератаки и модель киберугроз $G(V,E,A,O,P)$:

с) определить соответствующие значения вероятностей P_i^a, P_i^v, P_i^s;

д) вычислить $P_i = P_i^s + P_i^v - P_i^s P_i^v$ и $P_i = P_i^a(P_i^s + P_i^v - P_i^s P_i^v)$.

5. Определить множество путей кибератак для каждого из определенных объектов кибератак:

$$(A;o_1) = \{\{(a_1;o_1)\},\{(a_2;o_2)\},...,\{(a_K;o_1)\}\};$$

$$(A;o_2) = \{\{(a_1;o_2)\},\{(a_2;o_2)\},...,\{(a_K;o_2)\}\};$$

…

$$(A;o_M) = \{\{(a_1;o_M)\},\{(a_2;o_M)\},...,\{(a_K;o_M)\}\}.$$

6. Для каждого из объектов кибератак $o_i \in O$ на основе графа кибератак записать пути кибератаки (A,o_i) в виде функции алгебры логики (5.12):

$$y_i(Z_1,...,Z_N) = \bigvee_{l=1}^{d_i}\left[\bigwedge_{j\in K^i_{\phi l}} Z_j\right], i = \overline{1,M}.$$

7. Записать функцию опасного состояния (5.15) для ИКС:

$$y(Z_1,...,Z_N) = \bigvee_{i=1}^{M} y_i(Z_1,...,Z_N).$$

8. Преобразовать функцию опасного состояния в ОДНФ для построения функции вероятности успешности кибератаки.

9. Построить функцию вероятности успешности кибератаки (5.17) для ИКС, выполнив прямое замещение булевых переменных Z_i на их вероятностные значения $P\{Z_i = 1\} = P_i$:

$$P\{y(Z_1,...,Z_N) = 1\} \to P(P_1,...,P_N).$$

10. На основе значений вероятностей захвата объектов вычислить уровень защищенности ИКС.

5.10. Пример оценки уровня защищенности ИКС, структурной значимости ее объектов и анализ полученных результатов

5.10.1. Оценки уровня защищенности ИКС

Рассмотрим использование предложенной процедуры на примере стандартной локальной сети, физическая структура которой приведена на рис. 5.9. Данная ИКС представляет собой локальную сеть, которая через Межсетевой экран (Firewall) подключена к Интернет, содержит Веб-сервер (Web Server) и Почтовый сервер (Mail Server), которые связываются с Сервером баз данных (Database Server). Пользователи для работы используют Сервер приложений (Application Server), который тоже связывается с Сервером баз данных. Также в локальной сети присутствует Автоматизированное рабочее место (АРМ) администратора (Administrator), который имеет доступ ко всем серверам системы.

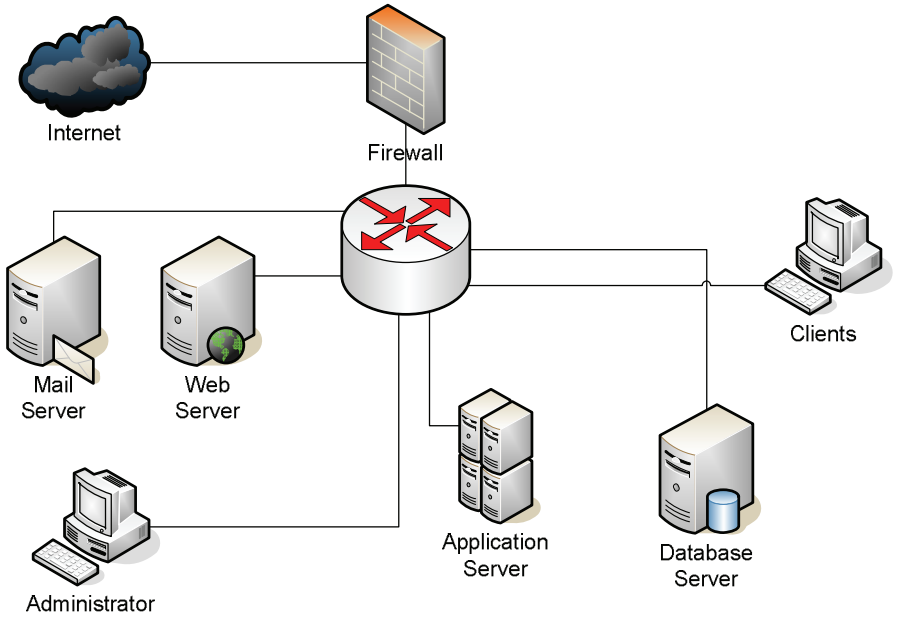


Рис. 5.9. Физическая структура логической сети

Рассмотрим случай, когда злоумышленник хочет получить доступ к серверу баз данных, и применим приведенную выше процедуру.

1. Имеем семь объектов информационно-коммуникационной системы:

v1 – Firewall;

v2 – Mail Server;

v3 – Web Server;

v4 – Administrator;

v5 – Clients;

v6 – Application Server;

v7 – Database Server.

2. Построим матрицу доступа объектов между собой (рис. 5.10), а также логическую схему локальной сети (рис. 5.11). В матрице доступности объектов показана связь объектов в компьютерной сети между собой. Если связь присутствует, то в столбце и строке, которые соответствуют порядковым номерам объектов, стоит 1; если связь отсутствует – 0. Например, в первой строке матрицы указано, что объект 1 (Firewall) имеет связь с объектами 2 (Mail Server) и 3 (Web Server).

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	1	0	0	0	0	0	1
3	1	0	0	0	0	0	1
4	1	1	1	0	1	1	1
5	0	0	0	0	0	1	0
6	0	0	0	0	1	0	1
7	0	0	0	0	0	1	0

1 – Firewall
2 – Mail Server
3 – Web Server
4 – Administrator
5 – Clients
6 – Application Server
7 – Database Server

Рис. 5.10. Матрица доступа объектов ИКС

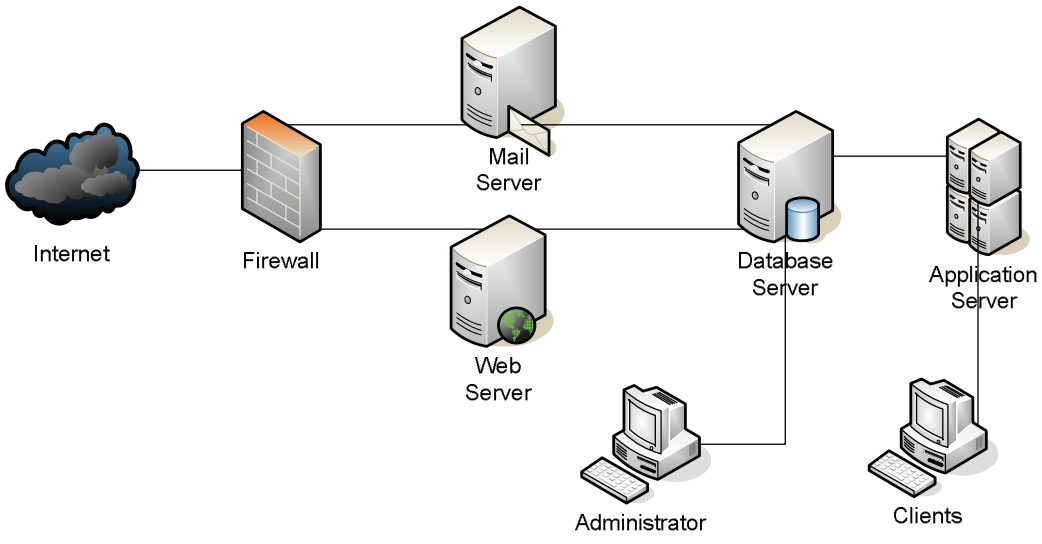


Рис. 5.11. Логическая структура локальной сети

3. Выделим две категории злоумышленников: внешние (из сети Интернет) и внутренние (из локальной сети). Таким образом, точками входа в систему будут

межсетевой экран, АРМ администратора и АРМ пользователя, значит источники кибератак будут следующие:

a1 = v1 – Firewall;

a2 = v4 – Administrator;

a3 = v5 – Clients.

В качестве объекта кибератаки для данного примера выберем Сервер баз данных: *o1 = v7 – Database Server.*

На основе матрицы доступа объектов и логической схемы локальной сети разработаем описательный сценарий развития опасного состояния, т. е. каким образом злоумышленник может проникнуть на сервер баз данных.

Это может случиться, если он имеет доступ к серверам, которые непосредственно имеют доступ к серверу БД. Такими являются Веб-сервер, Почтовый сервер, Сервер приложений и АРМ администратора. Если это внешний злоумышленник, то он для этого должен проникнуть через межсетевой экран и попасть на Веб-сервер или Почтовый сервер; если это внутренний злоумышленник, то он должен проникнуть на АРМ пользователя.

На основе матрицы доступа такие сценарии могут быть построены автоматически:

$$S_{1_1} = (v_1, v_2, v_7);\ S_{1_2} = (v_1, v_3, v_7);\ S_2 = (v_4, v_7);\ S_3 = (v_5, v_6, v_7).$$

Пути доступа злоумышленника к Серверу баз данных приведены на рис. 5.12.

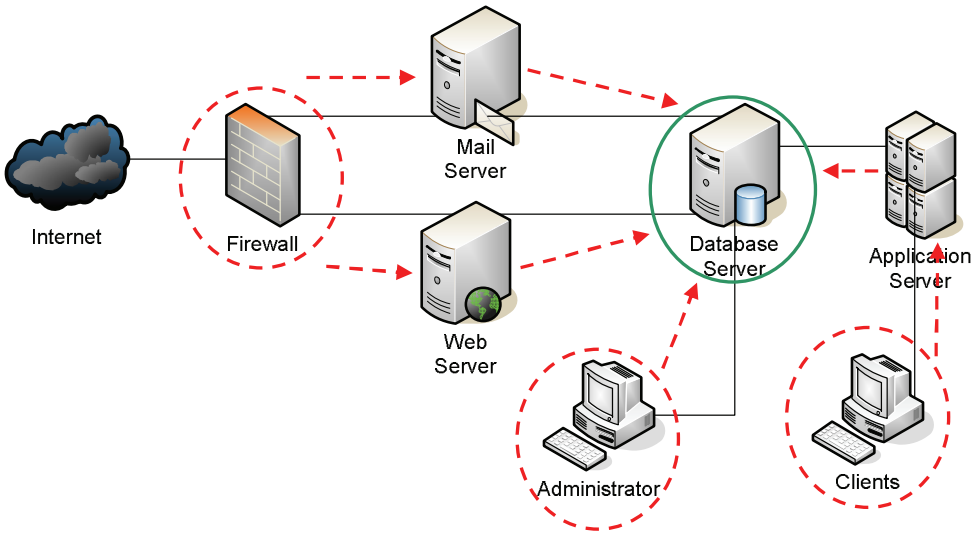


Рис. 5.12. Источники и пути кибератак на сервер управления базами данных

4. Представим сценарии захвата Сервера баз данных в виде графа кибератак с множеством соответствующих событий, которые приводят к этому (рис. 5.13).

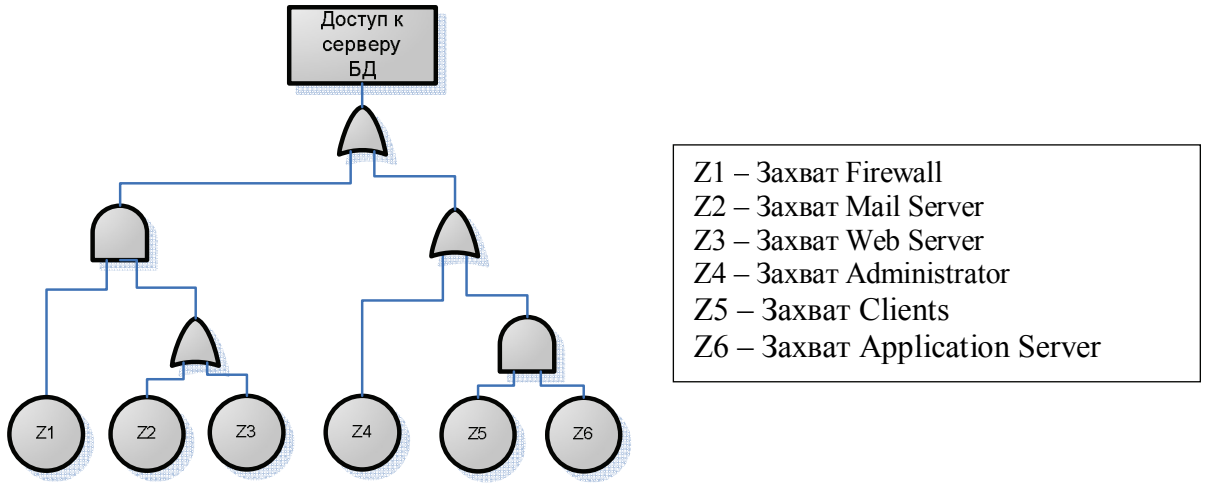


Рис. 5.13. Сценарий кибератак в виде графа атак

5. На основе указанного выше графа построим функцию опасного состояния в виде (5.12):

$$y(Z_1,...,Z_6) = Z_1(Z_2 \vee Z_3) \vee (Z_4 \vee Z_5 Z_6) = Z_1 Z_2 \vee Z_1 Z_3 \vee Z_4 \vee Z_5 Z_6 . \qquad (5.35)$$

6. Для перехода к вероятностной форме (5.17) запишем (5.35) в базисе конъюнкций-отрицаний:

$$y(Z_1,...,Z_6) = Z_1(Z_2 \vee Z_3) \vee (Z_4 \vee Z_5 Z_6) = Z_1(Z_2' Z_3')' \vee (Z_4'(Z_5 Z_6)')' =$$

$$= \left[\left(Z_1(Z_2' Z_3')' \right)' \left((Z_4'(Z_5 Z_6)')' \right)' \right]' = \left[\left(Z_1(Z_2' Z_3')' \right)' \left(Z_4'(Z_5 Z_6)' \right) \right]' . \qquad (5.36)$$

7. Запишем функцию вероятности успешности кибератаки $P\{y(Z_1,...,Z_6)=1\}$, заменив в (5.36) Z_i на $P\{Z_i = 1\} = P_i$:

$$P\{y(Z_1,...,Z_6)=1\} = P\left\{ \left[\left(Z_1(Z_2' Z_3')' \right)' \left(Z_4'(Z_5 Z_6)' \right) \right]' = 1 \right\} =$$

$$= 1 - \left[1 - P_1(1-(1-P_2)(1-P_3)) \right] \left[(1-P_4)(1-P_5 P_6) \right] = \qquad (5.37)$$

$$= P_4 + (1-P_4)P_5 P_6 + P_1 P_3 (1-P_4)(1-P_5) + P_1 P_3 (1-P_4) P_5 (1-P_6) +$$

$$+ P_1 P_2 (1-P_3)(1-P_4) P_5 (1-P_6) + P_1 P_2 (1-P_3)(1-P_4)(1-P_5).$$

Чтобы проверить полученное соотношение (5.37), воспользуемся граничными значениями, т. е. определим, при каких входных данных вероятность захвата сервера баз данных будет равна 1. В результате анализа получим табл. 5.2, которая соответствует сценарию кибератаки, приведенному в п. 3.

Таблица 5.2

Граничные вероятностные значения для захвата сервера

P_1	P_2	P_3	P_4	P_5	P_6	P
1	X	1	X	X	X	1
1	1	X	X	X	X	1
1	1	1	X	X	X	1
X	X	X	1	X	X	1
X	X	X	X	1	1	1

В строках табл. 5.2 приведены значения вероятностей, при которых общая функция вероятности будет равна 1 (при этом другие значения вероятностей могут быть любыми). Эти значения полностью совпадают с описательными сценариями реализации кибератак, что может служить подтверждением корректности модели и подхода.

Рассмотрим пример использования полученного соотношения (5.37) для оценивания вероятностей (обратной величине к уровню защищенности) успешности кибератаки на базу данных.

Пусть вероятностные значения условий, необходимых для совершения кибератаки, будут следующими (табл. 5.3).

В столбцах таблицы заданы модельные значения вероятностей реализации каждого из условий:

P_1 – захват Firewall;

P_2 – захват Mail Server;

P_3 – захват Web Server;

P_4 – захват АРМ-а Administrator;

P_5 – захват АРМ-а Clients;

P_6 – захват Application Server.

В последнем столбце P – уровень незащищенности Database Server, полученный путем подстановки значений $P_1, \ldots, P_6$ в соотношение (5.37).

Таблица 5.3

Вероятности захвата объектов

P_1	P_2	P_3	P_4	P_5	P_6	P
0,6	0,2	0,4	0,7	0,1	0,5	0,804
0,5	0,5	0,5	0,5	0,5	0,5	0,766

Как видно из табл. 5.3, изменения вероятностей захвата промежуточных объектов системы будут непосредственно влиять на результирующее значение вероятности успешной кибератаки на выбранный объект, в данном случае – сервер баз данных.

Построим таким же образом функции вероятности успешности кибератаки для Сервера приложений (v_6) и для Веб-сервера (v_3), а потом и для всей ИКС.

Функция вероятности успешности кибератаки для Сервера приложений:

1. Сценарий кибератаки.

Внутренний злоумышленник может непосредственно иметь доступ с АРМ пользователя и АРМ администратора.

Внешний злоумышленник может иметь доступ через межсетевой экран, далее через сервер электронной почты или веб-сервер, далее через сервер баз данных (рис. 5.14):

$$S_{1_1} = (v_1, v_2, v_7, v_6);\ S_{1_2} = (v_1, v_3, v_7, v_6);\ S_2 = (v_4, v_6);\ S_3 = (v_5, v_6).$$

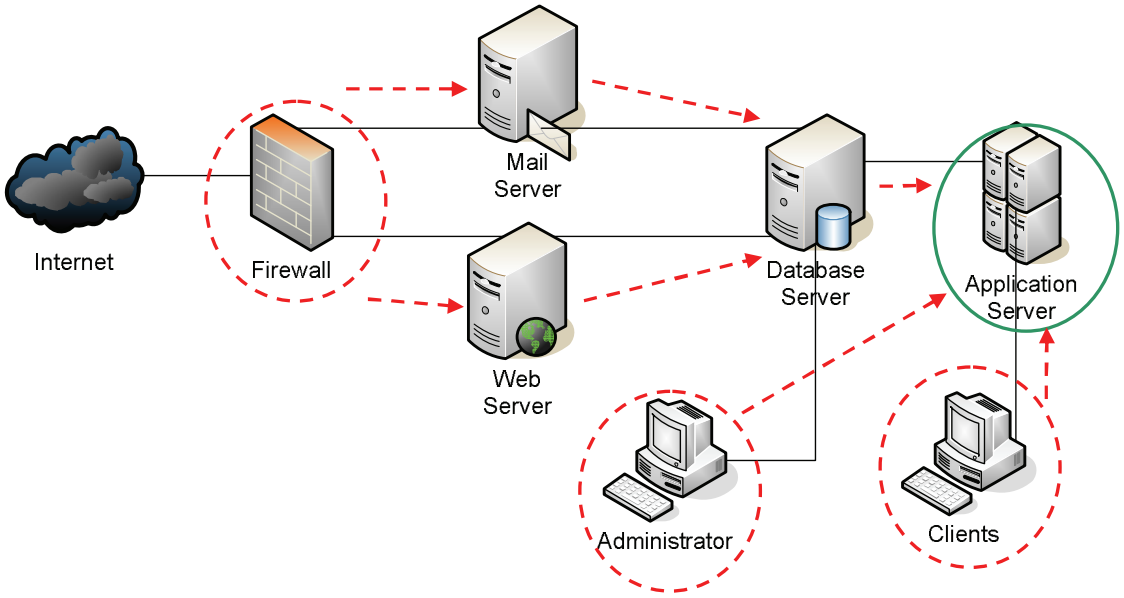


Рис. 5.14. Источники и пути кибератак на сервер приложений

2. Таким образом, граф кибератаки будет иметь следующий вид (рис. 5.15).

3. Функция опасного состояния имеет следующий вид:

$$y_2(Z_1, Z_2, Z_3, Z_4, Z_5, Z_7) = Z_1(Z_2 \vee Z_3)Z_7 \vee (Z_4 \vee Z_5) = Z_1Z_2Z_7 \vee Z_1Z_3Z_7 \vee Z_4 \vee Z_5.$$

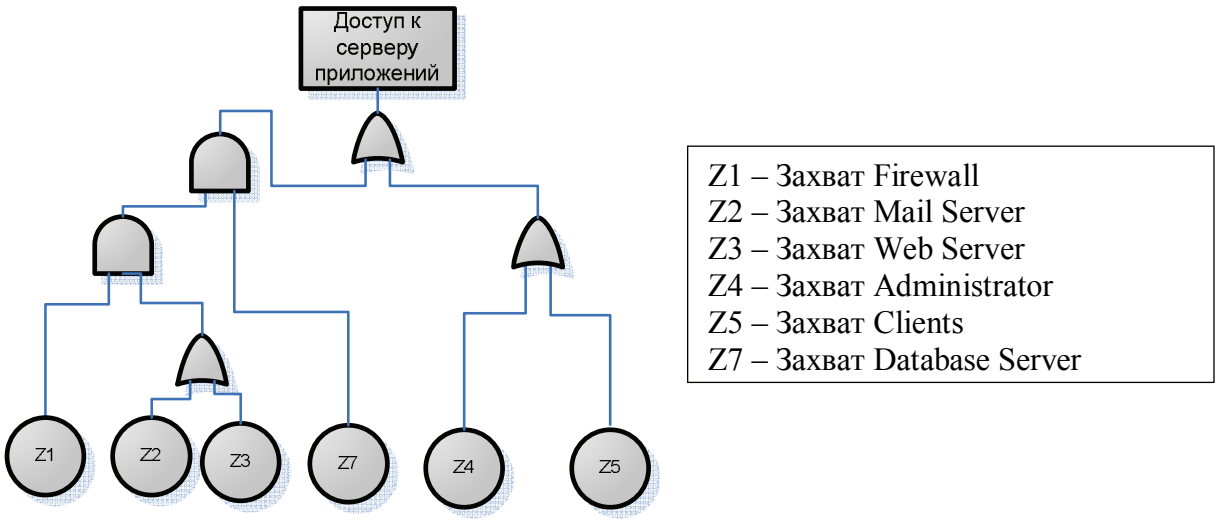


Рис. 5.15. Сценарий кибератак в виде графа атак

4. ОДНФ будет иметь следующий вид:

$$y_2(Z_1,Z_2,Z_3,Z_4,Z_5,Z_7) = Z_5 \vee Z_4 Z_5' \vee Z_1 Z_3 Z_4' Z_5' Z_7 \vee Z_1 Z_2 Z_3' Z_4' Z_5' Z_7.$$

5. Тогда функция вероятности успешности кибератаки будет иметь вид:

$$P_A\{y_2(Z_1,Z_2,Z_3,Z_4,Z_5,Z_7)=1\} = P_5 + P_4(1-P_5) + \\ + P_1P_3P_7(1-P_4)(1-P_5) + P_1P_2P_7(1-P_3)(1-P_4)(1-P_5). \quad (5.38)$$

Пусть вероятностные значения будут следующими (табл. 5.4).

Таблица 5.4

Вероятности захвата объектов

P_1	P_2	P_3	P_4	P_5	P_7	P
0,6	0,2	0,4	0,7	0,1	0,5	0,772
0,5	0,5	0,5	0,5	0,5	0,5	0,797

Функция вероятности успешности кибератаки для Веб-сервера:

1. Сценарий кибератаки.

Внутренний злоумышленник может иметь доступ непосредственно с АРМ администратора или с АРМ пользователя через сервер приложений и сервер баз данных.

Внешний злоумышленник может иметь доступ через межсетевой экран (рис. 5.16):

$$S_1 = (v_1, v_3);\ S_2 = (v_4, v_3);\ S_3 = (v_5, v_6, v_7, v_3).$$

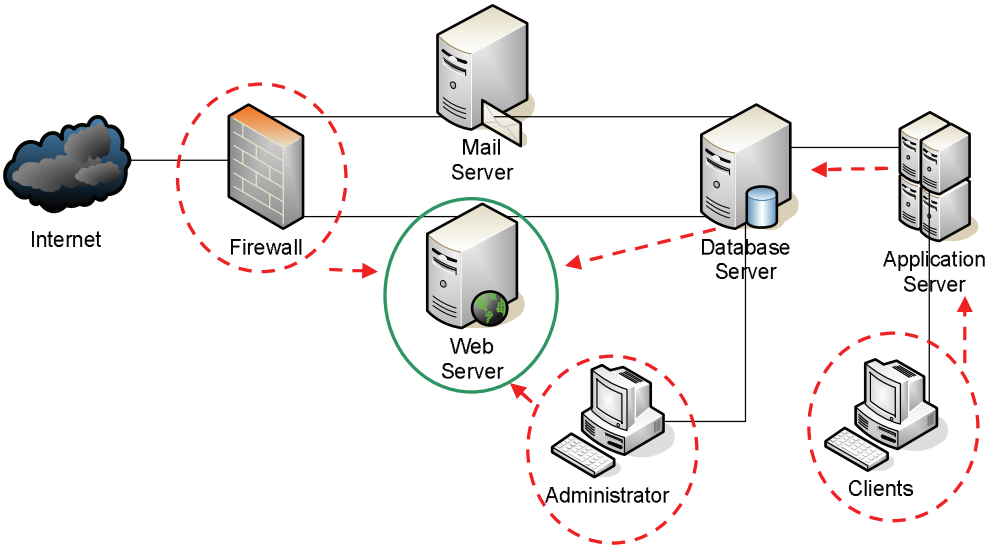


Рис. 5.16. Источники и пути атак на Веб-сервер

2. Граф кибератаки будет иметь следующий вид – рис. 5.17.

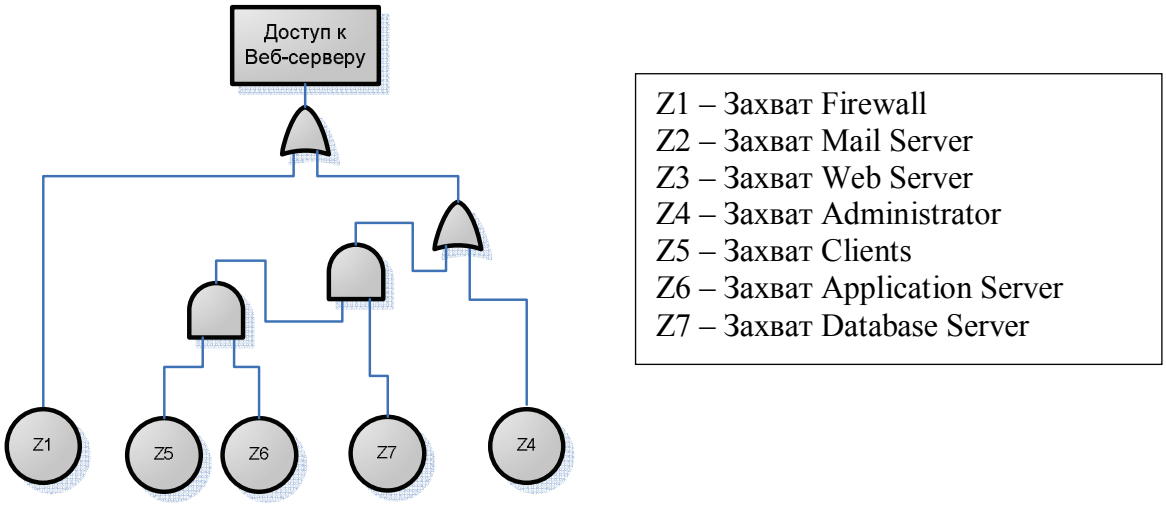


Рис. 5.17. Сценарий атак в виде графа атак

3. Функция опасного состояния будет иметь следующий вид:

$$y_3(Z_1, Z_4, Z_5, Z_6, Z_7) = Z_1 \vee Z_4 \vee Z_5 Z_6 Z_7.$$

4. В ОДНФ будет иметь следующий вид:

$$y_3(Z_1, Z_4, Z_5, Z_6, Z_7) = Z_4 \vee Z_1 Z_4' \vee Z_1' Z_4' Z_5 Z_6 Z_7.$$

5. Функция вероятности успешности кибератаки будет иметь вид:

$$P_W\{y_3(Z_1, Z_4, Z_5, Z_6, Z_7) = 1\} = P_4 + P_1(1 - P_4) + P_5 P_6 P_7 (1 - P_1)(1 - P_4). \qquad (5.39)$$

Пусть вероятности захвата будут следующими (такими же, как и в предыдущих таблицах) (табл. 5.5).

Таблица 5.5

Вероятности захвата объектов

P_1	P_2	P_3	P_4	P_5	P_7	P
0,6	0,2	0,4	0,7	0,1	0,5	0,883
0,5	0,5	0,5	0,5	0,5	0,5	0,781

Функция вероятности успешности кибератаки для ИКС. Для оценки уровня защищенности ИКС определим ключевые объекты системы. Система перейдет в опасное состояние, если в опасное состояние перейдет хотя бы один из ее ключевых объектов. К таким объектам отнесем Сервер баз данных – v_7, Сервер приложений – v_6 и Веб-сервер – v_3.

Функция опасного состояния для системы будет иметь следующий вид:

$$y(Z_1,...,Z_m) = y_1(Z_1,...,Z_7) \vee y_2(Z_1,...,Z_7) \vee y_3(Z_1,...,Z_7);$$

$$P\left\{y^S(Z_1,...,Z_m)=1\right\} = P\left\{\bigvee_{i=1}^{N} y_i(Z_1,...,Z_m)=1\right\} =$$

$$= P\left\{\left(y_1(Z_1,...,Z_7)=1\right) \vee \left(y_2(Z_1,...,Z_7)=1\right) \vee \left(y_3(Z_1,...,Z_7)=1\right)\right\}.$$

Граф кибератаки будет иметь вид, приведенный на рис. 5.18. Отметим, что он будет содержать одинаковые элементы и пути, которые повторяются.

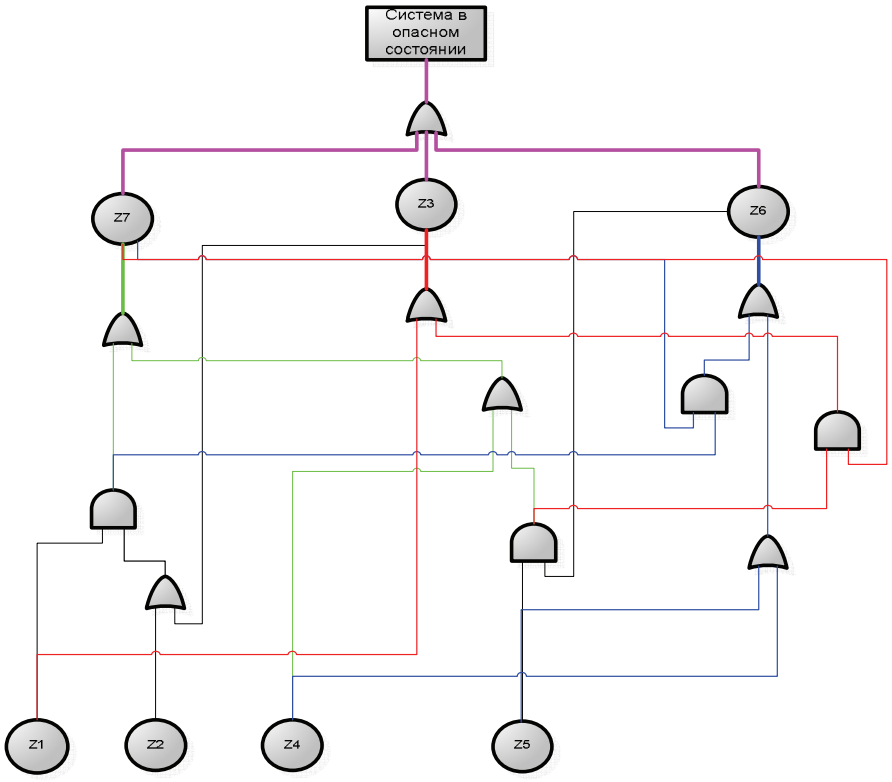


Рис. 5.18. Сценарий кибератак в виде графа атак для ИКС

Построим функцию вероятности успешной кибератаки:

$$P(P_1,P_2,P_3,P_4,P_5,P_6,P_7)=$$
$$=P_4P_7+1-P_4)P_5P_6+P_4P_5P_6(1-P_7)+P_4(1-P_5)P_6(1-P_7)+P_3P_4(1-P_5)(1-P_6)(1-P_7)+$$
$$+P_3P_4P_5(1-P_6)(1-P_7)+P_1P_3(1-P_4)(1-P_5)+P_1P_3(1-P_4)P_5(1-P_6)+ \quad (5.40)$$
$$+P_1P_2(1-P_3)(1-P_4)(1-P_5)P_7+P_1P_2(1-P_3)(1-P_4)P_5(1-P_6)P_7.$$

Для исследования полученной функции подставим значение вероятностей захвата объектов в (5.40) и проанализируем результирующие значения вероятности успешности кибератаки на ИКС. Подставим значения вероятностей из табл. 5.6.

Таблица 5.6

Вероятности захвата объектов ИКС

№	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P
1	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,68
2	0,5	0,5	0,5	0,5	0,5	0,5	0,7	0,71
3	0,5	0,7	0,5	0,5	0,5	0,5	0,5	0,69
4	0,7	0,5	0,5	0,5	0,5	0,5	0,5	0,73
5	0,5	0,5	0,5	0,5	0,7	0,5	0,5	0,71
6	1	0,5	1	0,5	0,5	0,5	0,5	1
7	0,6	0,2	0,4	0,7	0,1	0,5	0,5	0,69

В данной таблице в строках записаны разные возможные значения вероятностей захвата системы, а в последней колонке – вероятность успешности кибератаки на ИКС. В первой сроке табл. 5.6 рассматривается вариант, при котором вероятность захвата всех объектов равна 0,5, тогда вероятность успешности кибератаки будет равна 0,68.

В строках со второй по пятую увеличиваем значение вероятности захвата для одного из объектов до 0,7, оставляя все остальные вероятности равными 0,5. Анализируя при этом значения вероятностей успешности кибератак для каждой из строк, можно увидеть, что одинаковое увеличение вероятности захвата для объектов в разных строках будет по-разному влиять на результирующую вероятность защищенности. Например, хуже всего на вероятность повлияет увеличение вероятности захвата Firewall (4 строка, $P_1 = 0,7$). Из этого можно сделать вывод о том, что функция вероятности защищенности учитывает значимость разных объектов ИКС.

В шестой строке приведен пример того, что при значениях вероятностей захвата, равных 1 для всех объектов, которые входят в один из сценариев

кибератаки ($P_1 = P_3 = 1$), вероятность успешности кибератаки на ИКС также будет равна 1, т. е., как и было отмечено при построении модели, при успешной кибератаке на один из важных объектов системы она вся будет переходить в опасное состояние.

Таким образом, основываясь на результатах анализа, можно сделать вывод о том, что построенная функция вероятности успешности кибератаки (5.40) соответствует модели, а также адекватно отображает поведение информационно-коммуникационной системы с точки зрения практических соображений. Подробному исследованию зависимости функции вероятности успешности кибератаки от топологии сети будет посвящена следующая глава.

Полученные результаты могут быть применены для информационно-коммуникационных систем любого размера. Для большой системы вероятностная функция может содержать много переменных и иметь сложный вид, поэтому необходима автоматизация процесса построения вероятностной функции и вычисления ее значения.

5.10.2. Оценка структурной значимости объектов ИКС

Определим структурную значимость объектов для примера сети из предыдущего пункта. Для этого воспользуемся полученными функциями вероятности успешности кибератаки для отдельных объектов системы и всей системы в целом:

P_D – функция вероятности успешности кибератаки Сервера баз данных (5.37);

P_A – функция вероятности успешности кибератаки Сервера приложений (5.38);

P_W – функция вероятности успешности кибератаки Веб-сервера (5.39);

P – функция вероятности успешности кибератаки ИКС (5.40).

Для примера приведем вычисления структурной значимости по функции (5.40) по параметру P_3, подставив $P = P_1,...,P_N$, а потом вместо P – значение 0,5:

$$\xi_3 = \frac{\partial P(P_1,...,P_7)}{\partial P_3} =$$

$$= [P_4(1-P_5)(1-P_6)(1-P_7) + P_4P_5(1-P_6)(1-P_7) + P_1(1-P_4)(1-P_5) + P_1(1-P_4)P_5(1-P_6)] -$$

$$-[P_1P_2(1-P_4)(1-P_5)P_7 + P_1P_2(1-P_4)P_5(1-P_6)P_7] =$$

$$= P^3 + 3P^4 - P^5 - P^6 = (0,5)^3 + 3(0,5)^4 - (0,5)^5 - (0,5)^6 \approx 0,266.$$

Таким же образом вычисляем и другие значения структурной значимости объектов системы. Полученные результаты находятся в табл. 5.7.

Таблица 5.7

Значимость объектов ИКС

Объекты	Значимость объектов			
	ξ_D	ξ_A	ξ_W	ξ
ξ_1	0,281	0,094	0,438	0,234
ξ_2	0,094	0,031	0,000	0,047
ξ_3	0,094	0,031	0,000	0,266
ξ_4	0,469	0,406	0,438	0,391
ξ_5	0,156	0,406	0,063	0,172
ξ_6	0,156	0,000	0,063	0,297
ξ_7	0,000	0,094	0,063	0,172

Z1 – Захват Firewall
Z2 – Захват Mail Server
Z3 – Захват Web Server
Z4 – Захват Administrator
Z5 – Захват Clients
Z6 – Захват Application Server
Z7 – Захват Database Server

В таблице приведены значения структурной значимости для всех семи объектов $(v_1,...,v_7)$, которые присутствуют в компьютерной сети, при этом также вычислена значимость объектов по каждой из функций вероятностей успешности атаки $P_D,\ P_A,\ P_W,\ P$.

По таблице построены два графика: на первом (рис. 5.19) показана структурная значимость каждого из объектов по каждой функции вероятностей успешности кибератаки; на втором графике (рис. 5.20) приведены значения, полученные только на основе функции вероятности успешности кибератаки для ИКС.

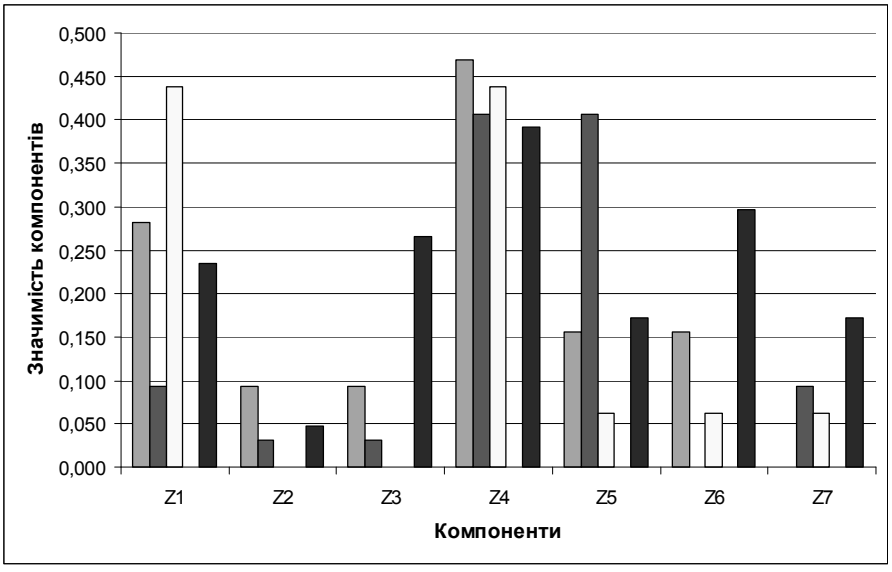


Рис. 5.19. Значимость объектов ИКС

Анализируя таблицу и график, можно увидеть, что в зависимости от того, по какой из функций вероятности успешности кибератаки определяется структурная значимость объектов, результаты будут частично разными. И эти результаты будут справедливыми, так как с точки зрения каждого из объектов их защищенность будет зависеть от вероятностей захвата только этих объектов, с которыми они непосредственно взаимодействуют.

Для функций вероятностей успешности кибератаки, построенных для критических объектов системы, наиболее структурно значимыми будут:

– ξ_D – захват АРМ администратора (Z_4) и межсетевого экрана (Z_1);

– ξ_A – захват АРМ администратора (Z_4) и АРМ пользователя (Z_5);

– ξ_W – захват АРМ администратора (Z_4) и межсетевого экрана (Z_1).

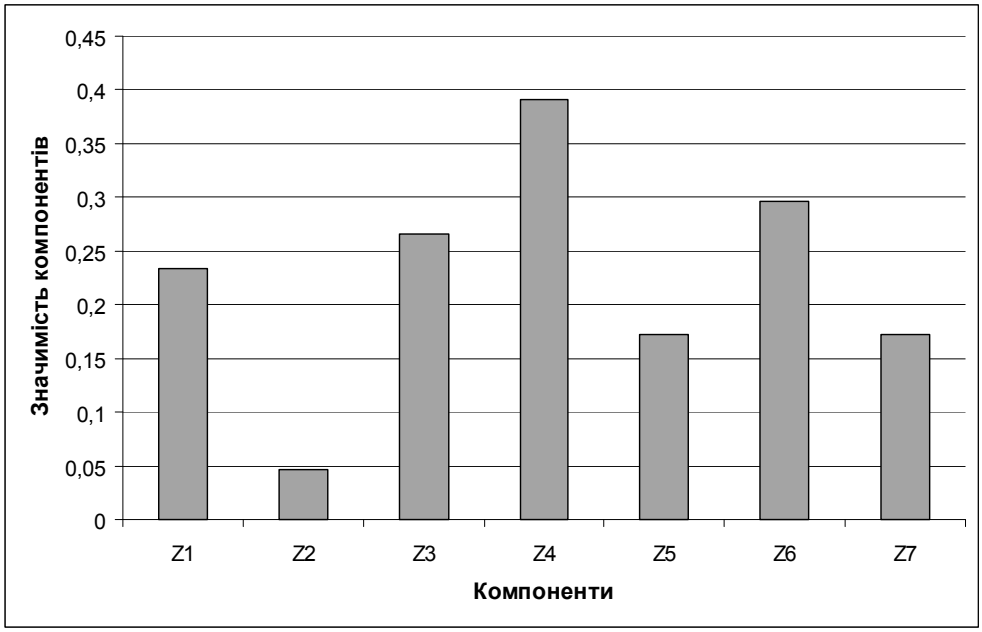


Рис. 5.20. Структурная значимость объектов ИКС

Для функции вероятности успешности кибератаки для ИКС (ξ) наиболее структурно значимыми объектами будут (рис. 5.20):

– захват АРМ администратора (Z_4);

– захват Сервера приложений(Z_6);

– захват Веб-сервера (Z_3);

– захват межсетевого экрана (Z_1).

Как можно увидеть, для всех функций вероятности наиболее структурно значимым объектом системы будет АРМ администратора (Z_4). Это объясняется

тем, что с этого объекта существует непосредственный доступ ко всем другим объектам, т. е. захват АРМ администратора будет самым коротким путем осуществления кибератаки на другие объекты системы.

Сервер приложений (Z_6) является структурно значимым для безопасности системы, так как через него проходит много информационных потоков и путей кибератак, а через Веб-сервер (Z_3) и межсетевой экран (Z_1) будут проходить кибератаки из Интернет.

Основываясь на этих вычислениях еще на этапе проектирования ИКС, можно сделать вывод о том, что необходимо уделить больше внимания механизмам защиты соответствующих объектов системы или изменить структуру (топологию) сети для уменьшения структурной значимости объектов.

Особенно это касается АРМ администратора. Например, для уменьшения вероятности его захвата можно ввести ограниченный доступ к компьютеру, повысить требования к политике паролей, ввести дополнительные электронные средства идентификации и аутентификации, но при этом значимость этого сервера не уменьшится. Причина в том, что с этого сервера существует доступ ко всем объектам сети. И единственным вариантом для уменьшения значимости является изменение структуры самой сети.

Также следует отметить, что значения структурной значимости объектов зависят только от топологии сети и разработанных сценариев атак. При подстановке в функцию чувствительности реальных значений вероятностей захвата объектов результаты могут быть существенно другими, поэтому при проектировании ИКС на первом этапе можно вычислить структурную значимость объектов и учесть эти значения при проектировании системы защиты. А на втором этапе, когда станут известны вероятности захвата объектов, произвести вычисления значимости объектов заново и в дальнейшем основываться на этих данных.

Выводы

В данной главе описана модель ИКС, которая учитывает топологию сети ИКС, и на основе которой построена комплексная модель киберугроз в виде графа кибератак.

Используя логико-вероятностный метод, построена функция вероятности успешности кибератаки, исследованы ее структура и свойства. На ее ос-

нове определена функция чувствительности для определения структурной значимости объектов ИКС.

Разработана процедура построения функции вероятности успешности атаки и приведены примеры для вычисления уровня защищенности модельной ИКС. Показано, что структура и значение функции вероятности успешности атаки будут зависеть от топологии сети ИКС, источников и объектов кибератак, а также от вероятностей захвата объектов системы.

6. ПОСТРОЕНИЕ БЕЗОПАСНОЙ ТОПОЛОГИИ СЕТИ ИКС НА ОСНОВЕ СТРУКТУРНОЙ ЛОГИКО-ВЕРОЯТНОСТНОЙ МОДЕЛИ

В данной главе рассматриваются вопросы построения безопасной топологии сети ИКС на основе структурной логико-вероятностной модели, показано влияние топологии сети на уровень защищенности ИКС, предложен подход к построению безопасной топологии сети.

Исходя из того, что функция вероятности успешности кибератаки, рассмотренная в предыдущей главе, зависит от топологии сети, будем использовать ее значение в качестве критерия для выбора безопасной топологии. Будем считать, что некая топология сети будет более безопасной по сравнению с другими вариантами топологий, если функция вероятности успешности кибератаки принимает на ней минимальное значение.

6.1. Анализ влияния топологии сети на уровень защищенности ИКС

Как было показано в предыдущей главе, топология сети ИКС – это конфигурация графа, вершинам которого отвечают узлы сети – объекты/серверы/компьютеры/коммутационные устройства, а ребрам – связи между ними. Также, напомним, что ИКС состоит из набора определенных сервисов, которые через информационные или технологические потоки могут зависеть один от другого.

Топология сети, в первую очередь, будет формироваться, исходя из функциональных нужд организации, а также учитывая зависимость между объектами (сервисами) системы. Кроме этого, может быть набор дополнительных условий, связанных с территориальным размещением объектов, структурой организации, политикой безопасности и прочее.

Но, даже с учетом всех этих условий, во время проектирования топологии сети, может возникать несколько ее вариантов. Эти варианты могут быть одинаковыми с точки зрения бизнес-функциональности системы, но оказаться существенно разными с точки зрения кибербезопасности ИКС.

Если взять две информационно-коммуникационные системы, которые представлены графами $G_1 = G(V, E_1)$ и $G_2 = G(V, E_2)$, с одинаковым набором

объектов $V = (v_1,...,v_N)$, но которые имеют разные топологии E_1 и E_2, то одна из этих систем может быть безопаснее, чем другая. Покажем это на примере двух сетей, логические схемы которых приведены на рис. 6.1, *а*, *б*.

Топология сети G_1 на рис. 6.1, *а* является одноранговой (полностью связной), т. е. каждый из объектов имеет доступ ко всем другим объектам. На рис. 6.1, *б* в сети G_2 есть лишь те связи между ее объектами, которые необходимы для выполнения системой своих функций.

Исходя из практических соображений, а также рекомендаций по безопасности при проектировании топологии сети, можно сделать вывод, что сеть G_2 будет более защищенной, чем G_1. Покажем это. Матрицы доступности объектов для этих сетей будут следующими (рис. 6.1).

$$G_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad G_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

1 – Firewall
2 – SMTP Server
3 – Web Server
4 – Clients
5 – Application Server
6 – Database Server

Рис 6.1. Матрицы доступности объектов ИКС

Как можно видеть, сеть G_2 является частичным случаем сети G_1, из которой был убран ряд связей между объектами.

Для формального сравнения уровня защищенности сетей G_1 и G_2 воспользуемся функцией вероятности успешности кибератак $P(G, A, O, P_1,..., P_N)$, которая была получена в гл. 5. Построим графы кибератак $G(V, E, A, O)$ для двух сетей.

Предположив, что источники угроз $A \subset V$ и объекты $O \subset V$ кибератак будут одинаковыми для двух сетей, в результате чего для каждой из них получим функции вероятности успешности кибератак, которые зависят только от топологий: $P(G_1)$ и $P(G_2)$. Исходя из того, что сеть G_1 является менее защищенной, при сравнении значений этих функций должно всегда выполняться неравенство $P(G_1) \geq P(G_2)$.

Источники угроз, объекты кибератак и пути (сценарии) кибератак приведены на рис. 6.2. Основываясь на этом и пользуясь алгоритмом, приведенным

в гл. 5, запишем функции вероятности успешной кибератаки для каждой из сетей:

$$P(G_1) = P_1P_3 + (1-P_1)P_3P_4 + P_1(1-P_3)P_6 + (1-P_1)(1-P_3)P_4P_6;$$

$$P(G_2) = P_1P_3 + (1-P_1)P_4P_5P_6 + P_1(1-P_3)P_4P_5P_6 + P_1P_2(1-P_3)(1-P_4)P_5P_6.$$

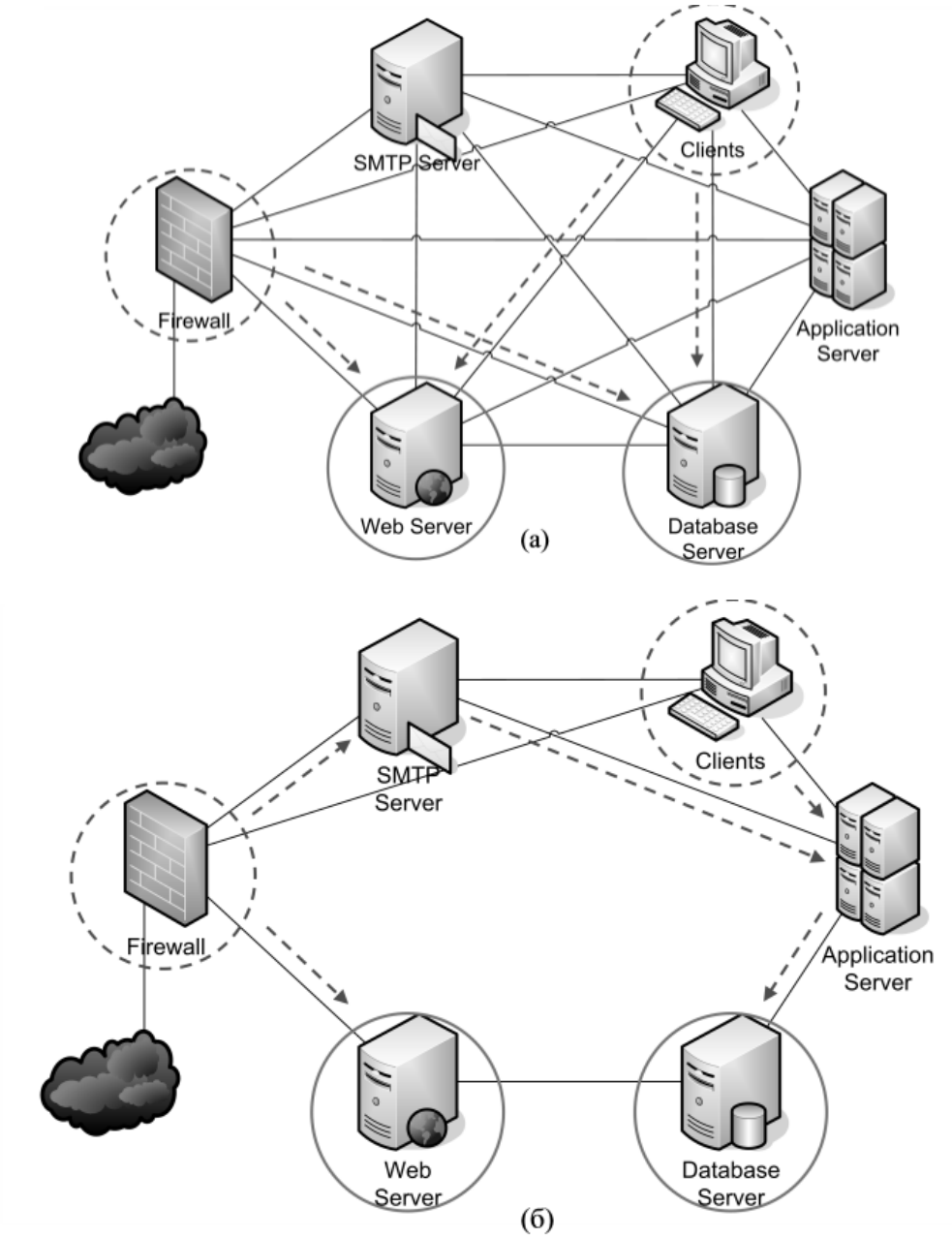


Рис. 6.2. Сети, которые имеют одинаковую функциональность, но разные топологии

Теперь необходимо сравнить значения этих функций. Исходя из того, что они являются функциями многих переменных (часть из которых являются разными) и имеют разную структуру, то в общем случае такое сравнение сделать довольно тяжело, поэтому, для упрощения данного примера,

предположим равенство всех вероятностей захвата объектов $P = P_1 = ... = P_N$. После этого получим функции следующего вида:

$$P(G_1) = P^4 - 4P^3 + 4P^2;$$

$$P(G_2) = P^6 - 3P^5 + P^4 + P^3 + P^2.$$

Покажем, что $\forall P \in [0,1] : P(G_1) \geq P(G_2)$, и равенство достигается лишь на краях промежутка. Чтобы показать это, построим графики (рис. 6.3) указанных функций.

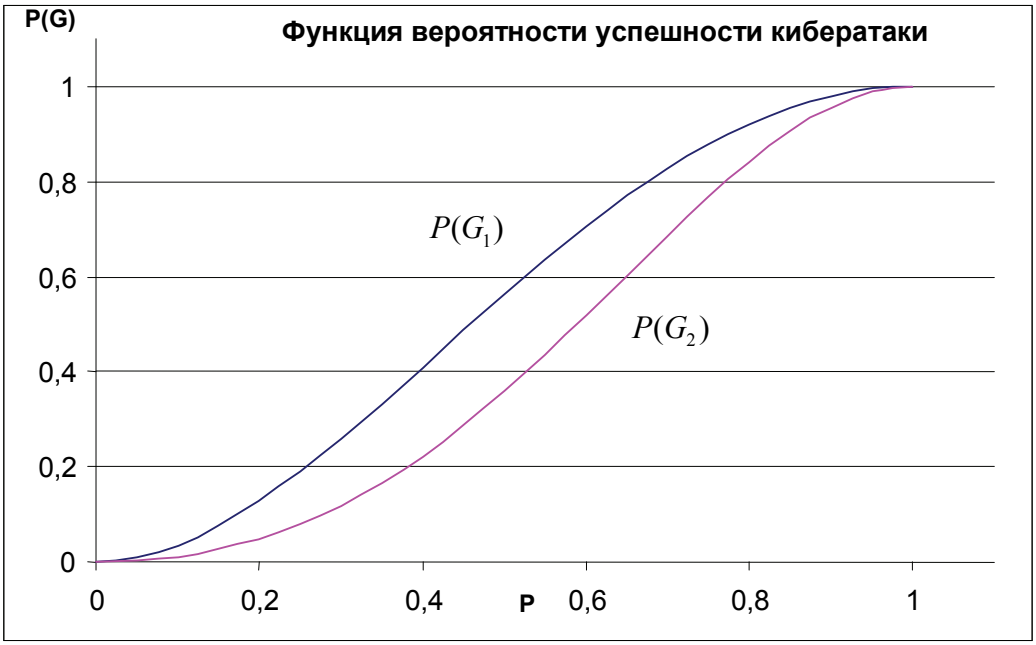


Рис. 6.3. Графики функций вероятности успешности атаки

Из графика (рис. 6.3) видно, что $P(G_1) \geq P(G_2)$, т. е. вероятность успешности атаки на сеть G_1 будет всегда больше, чем на сеть G_2, при одинаковых вероятностях захвата объектов сети.

Таким образом, для частичного случая с помощью функций вероятности успешности атаки было показано, что топология сети будет влиять на ее уровень защищенности. И на этапе проектирования топологии на это нужно обращать внимание.

Кроме этого, выводы о влиянии топологии сети на уровень защищенности ИКС можно сделать, исходя из анализа многих рекомендаций, руководств, инструкций, практических советов от известных производителей программных и аппаратных средств. В частности, в этих рекомендациях описаны стандартные решения по проектированию топологии сетей с учетом требований к защите

информации, в зависимости от размера ИКС, их функционального назначения, территориального размещения, категорий информации и др.

Исходя из этого, предложим формальную процедуру, с помощью которой можно было бы выполнять построение безопасной топологии сети ИКС.

6.2. Формальная постановка задачи построения безопасной топологии ИКС

Как было показано, топологию сети ИКС можно представить в виде графа $G(V,E)$, где V – множество сервисов сети; E – наличие связи между ними, $E \subseteq V \times V$. Рассмотрим общие задачи, которые могут быть связаны с проектированием и оптимизацией топологических/сетевых структур.

Согласно [140], формально задача топологического проектирования сетей сводится к поиску минимума некоторого функционала приведенной стоимости

$$F(U,\Omega,Y) \to \min$$

при наличии ограничений на вероятностно-временные и структурные характеристики сети

$$V_i(U,\Omega,Y) \le V_{i0},$$

а также требований принадлежности множества вариантов архитектуры сети $Q(U,\Omega,Y)$ к технически реализуемым решениям

$$Q(U,\Omega,Y) \in Q_0.$$

Можно выделить четыре типа задач, связанных с оптимизацией сетевых структур [141]:

Тип 1: Оптимизация структуры.

Найти граф $G(V,E)$, где V – множество узлов; E – множество ребер, которые обеспечивают экстремум некоторого структурного (целевого) функционала $F[G]$.

Тип 2: Динамическая оптимизация на статическом графе.

Для заданного графа $G(V,E)$ и динамической системы Φ на G,

$$\Phi(x,\dot{x},...,\{\alpha\},t) = 0,$$

найти значение параметров $\{\alpha\}$, которые обеспечивают экстремум глобального функционала $F[\Phi]$, динамической системы Φ. Переменные x – величины, которые представляют свойства узлов и ребер в сетевой структуре.

Тип 3: Оптимизация структуры для динамической системы.

Для заданной динамической системы Φ и множества параметров $\{\alpha\}$ найти граф $G(V,E)$, для которого глобальный функционал $F[\Phi]$ динамической системы Φ будет принимать экстремум.

Тип 4: Оптимизация для сетевой структуры, которая динамично изменяется.

Если граф сетевой структуры изменяется во времени, т. е. $G(V,E) = G(V,E,t)$, независимо или в связи с динамической системой Φ, необходимо найти значения параметров $\{\alpha\}$, для которых глобальный функционал $F[G,\Phi]$ динамической системы Φ и графа будет иметь экстремум.

Нужно также отметить работы Ю. П. Зайченко, например [142], в которых предложено много эффективных алгоритмов и решений практических задач построения компьютерных сетей.

Задачу построения безопасной топологии сети ИКС будем рассматривать как задачу Типа 1. Для ее решения необходимо определить функционал $F[G]$, с помощью которого можно сравнивать разные варианты топологий сетей, а также должны учитываться функциональные требования и ограничения на приемлемые варианты топологий.

В данном случае учет функциональных требований и ограничений является принципиальным, так как с точки зрения информационной безопасности наилучшим вариантом топологии сети будет тот, когда все объекты не будут связаны между собой, а самым худшим – когда все объекты будут иметь связь между собой (случай одноранговой сети).

Если рассматривать задачу поиска безопасной топологии в общем случае, для всех возможных вариантов топологии Ω (без учета дополнительных условий и ограничений), то для сети, которая содержит N объектов, возможное количество вариантов топологий ($|\Omega|$) будет равно

$$|\Omega| = 2^{N(N-1)} \tag{6.1}$$

в случае, когда учитываются направления связей между объектами, и

$$|\Omega| = 2^{N(N-1)/2}, \tag{6.2}$$

когда направления связей не учитываются.

Для примера простая сеть, которая приведена на рис. 6.2, *а*, *б*, и которая содержит 6 объектов ($N = 6$), может иметь 2^{30} или 2^{15} вариантов топологии.

Но не все из этих топологий будут приемлемы с точки зрения функционирования сети, и благодаря этому их общее количество может быть существенно уменьшено. Приемлемые варианты топологии могут определяться на основе следующих условий и ограничений:

– требуемой функциональности;
– зависимости сервисов один от другого;
– территориального размещения объектов;
– организационной структуры;
– пропускной способности;
– политики безопасности, принятой в ИКС.

Итак, с учетом приведенных условий и ограничений, получим некое множество $O(G) = \{G_1, G_2, ..., G_k\} \subseteq \Omega$, где G_i – допустимые варианты топологии сети.

Таким образом, общая задача построения безопасной топологии сети ИКС будет формулироваться следующим образом: среди приемлемых вариантов топологий найти топологию сети (структуру графа), которая обеспечивает экстремум функционала защищенности сети.

Формализованный вид задачи будет следующий:

$$\begin{cases} G^* = \arg\max\limits_{G \in O} F[G] \\ O(G) = \{G_1, G_2, ..., G_k\} \subseteq \Omega \end{cases}, \tag{6.3}$$

где $F[G]$ – функционал защищенности сети;

$O(G)$ – множество допустимых вариантов топологии сети;

Ω – все возможные варианты топологии сети.

В следующем пункте сформулируем свойства, которые должны быть у функционала защищенности, а также покажем, что в качестве этого функционала может быть использована функция вероятности успешности кибератаки.

6.3. Использование функции вероятности успешности кибератаки для построения безопасной топологии

Для начала, исходя из опыта и практических соображений, сформулируем основные свойства, которые должен иметь функционал защищенности:

– зависимость от топологии – при изменении топологии сети должно изменяться значение функционала защищенности;

– предельные значения – значение функционала для одноранговой (полностью связанной) сети должно быть наихудшим, а для полностью несвязанной – наилучшим;

– монотонность – значение функционала не возрастает при удалении одного или нескольких объектов, или связей между объектами.

В гл. 5 была предложена функция для оценки вероятности успешности кибератаки на ИКС. Учитывая то, что с помощью этой функции можно определять уровень защищенности ИКС, и то, что она зависит от топологии сети ИКС, возьмем ее в качестве функционала защищенности сети.

Как было сказано, функционал защищенности сети должен иметь основные три свойства. Покажем, что функция вероятности успешности кибератаки обладает указанными свойствами.

Зависимость от топологии

Как было показано, в общем случае, функция вероятности $P(G, A, O, P)$ имеет переменную структуру, которая определяется на основе графа кибератак $G(V, E, A, O)$:

– топологии сети ИКС – $G(V, E)$;

– источников угроз – A, и объектов – O кибератак.

Это обозначает, что изменение параметров G, A, O будет приводить к изменению ее структуры, которая будет также влиять на значение вероятности успешности кибератаки. Причем, если источники угроз и объекты кибератак находятся за рамками ИКС, хоть и зависят от назначения системы и ее свойств, то топология сети будет целиком зависеть от разработчиков системы. Поэтому, при проектировании топологии сети, объекты кибератак и источники угроз будут считаться фиксированными, т. е. на данном этапе будем считать, что структура функции вероятности успешности кибератаки будет зависеть лишь от топологии сети G. Значения вероятностей захвата объектов будем считать также фиксированными, поэтому функцию вероятности успешности кибератаки можно представить как

$$P(G) = P(G, A, O, P). \qquad (6.4)$$

Граничные значения

Рассмотрим значения функции вероятности успешности кибератаки для предельных вариантов топологии сети: полностью не связанной и полностью связанной.

При полностью не связанном графе сети (пример приведен на рис. 6.4, *а*) функция вероятности успешности кибератаки будет $P(G) = 0$. Это следует из того, что для такого случая невозможно записать ни одного из сценариев атак, кроме прямых атак.

При отсутствии связей между объектами системы ее защищенность будет определяться исключительно степенью защищенности ее отдельных объектов, а также физическими и организационными средствами защиты. Согласно НД ТЗИ такой вариант ИКС подпадает под определение АС класса «1», который не рассматривается в данной работе.

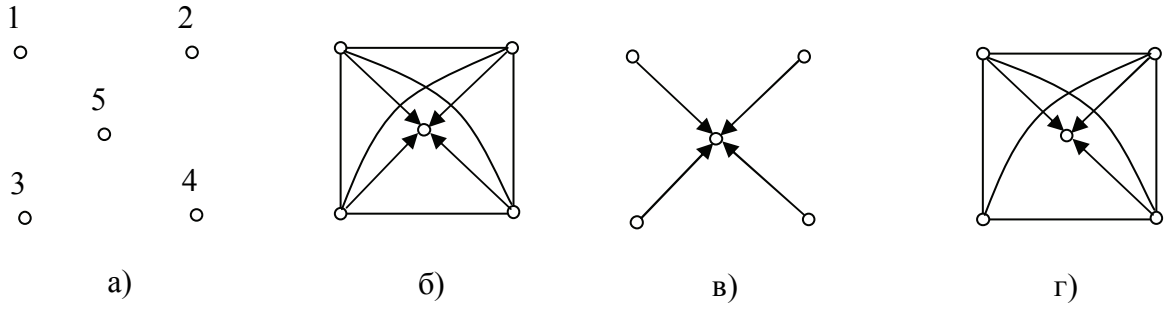


Рис. 6.4. Варианты связей между объектами ИКС

При полностью связанном графе сети (пример приведен на рис. 6.4, *б*) функция вероятности успешности кибератаки для одного объекта системы будет иметь вид:

$$P(G) = P((Z_1 \vee Z_2 \vee Z_3 \vee Z_4) = 1) =$$

$$= P_1 + (1 - P_1)P_2 + (1 - P_1)(1 - P_2)P + (1 - P_1)(1 - P_2)(1 - P_3)P_4 .$$

Такой вид функции обусловливается тем, что, согласно правилам алгебры логики, функция опасного состояния строится как кратчайший путь осуществления атаки, т. е., если взять источником угроз вершину А, а объектом – вершину В, как на рис. 6.4, *б*, то путем атаки будет путь АС (рис. 6.5).

Это вытекает из того, что функция опасного состояния для вершины С будет выглядеть следующим образом:

$$y(Z_1, Z_2) = Z_1 \vee (Z_1 \wedge Z_2) = Z_1 ,$$

где события Z_1 и Z_2 определяют вероятности успешного захвата вершин А и В, соответственно.

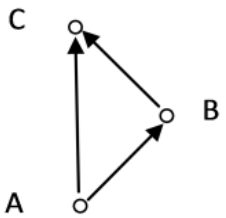


Рис. 6.5. Пути атак

Исходя из этого, граф, приведенный на рис. 6.4, *б*, будет иметь одну и ту же функцию опасного состояния, что и граф на рис. 6.4, *в*,

т. е. для полностью связанного графа сети (одноранговой сети) функция вероятности успешности атаки будет зависеть лишь от количества источников угроз и вероятностей их захвата:

$$P(G) = P\{(Z_1 \vee Z_2 \vee Z_3 \vee ... \vee Z_M) = 1\} =$$
$$= P(P_1,...,P_M) = \sum_{i=1}^{n} P_i - \sum_{i<j} P_i P_j + \sum_{i<j<k} P_i P_j P_k - ... + (-1)^{M-1} P_1 ... P_M , \quad (6.5)$$

где M – количество источников угроз.

Если предположить, что все вероятности захвата объектов в (6.5) равны $P_i = P$, то получим следующее выражение:

$$P(G) = M \cdot P - C_M^2 P^2 + C_M^3 P^3 - ... + (-1)^M P^M = M \cdot P + \sum_{l=2}^{M} (-1)^{l+1} C_M^l P^l . \quad (6.6)$$

В зависимости от значений вероятности P и от количества источников угроз M, функция вероятности успешности кибератаки (6.6) будет принимать вид, приведенный на графике (рис. 6.6).

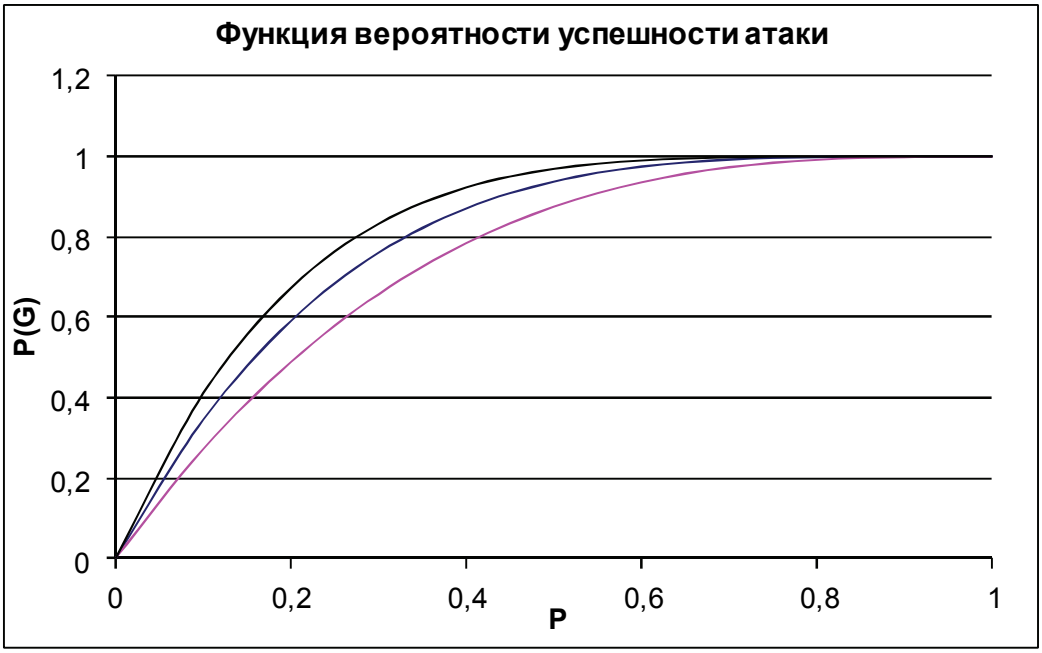


Рис. 6.6. Значение функции вероятности успешности атаки

Чем больше количество источников угроз, тем более крутым будет вид графика. Также, исходя из (6.6), можно увидеть, что вероятность успешности кибератаки на ИКС будет гарантированно выше максимального из значений вероятности захвата объекта системы:

$$\max(P_i) \le P(G) \le 1 .$$

Утверждение 3. Функция вероятности успешности кибератаки для полностью связанной топологии (одноранговой) сети будет не меньшей, чем все другие функции вероятностей успешности кибератаки, построенные для других вариантов топологии этой сети:

$$\forall G_i \in \Omega : P(G_i) \le P(G_{\max}),$$

где $G_{\max}$ – полностью связанная топология;

Ω – все возможные варианты топологии.

Справедливость данного утверждения будет вытекать из свойства монотонности функции вероятности успешности кибератаки, которое заключается в том, что если $G_1 \subset G_2$, то $P(G_1) \le P(G_2)$. Свойство монотонности функции вероятности было доказано в утверждении 1, в предыдущей главе.

Если $G_2 = G_{\max}$ (полностью связанный граф), то все другие графы могут быть получены путем исключения ребер из максимального графа, и соответственно всегда будет выполняться $\forall G_1 \in \Omega : G_i \subseteq G_{\max}$. Из этого будет следовать, что $P(G_i) \le P(G_{\max})$.

Монотонность

Следующее свойство, которое должен иметь функционал защищенности сети, – это монотонность.

Следствием удаления связей между объектами ИКС является то, что через это ребро перестанет проходить путь (сценарий) кибератаки, что, в свою очередь, будет приводить к уменьшению поверхности атак на ИКС.

Проанализируем детально, с использованием разработанной модели, что происходит со сценарием кибератак и, соответственно, с функцией вероятности успешности кибератаки при удалении связи между объектами сети.

Возможны следующие варианты:

– сценарии кибератак не изменяются – дерево кибератак не изменяется;

– из сценария кибератак полностью исчезает один или несколько путей атак;

– в сценарии кибератак появляется более длинный путь атаки.

Рассмотрим эти варианты подробнее:

1. Сценарии кибератак не изменяются.

Это значит, что через изъятое ребро графа сети не проходили возможные пути атаки. В этом случае не будет изменяться вид функции опасного состояния системы и функции вероятности успешности кибератаки.

2. Из сценария кибератак полностью исчезает один или несколько путей атак, возможно вместе с источником или источниками угроз.

Это может произойти, если через изъятое ребро графа сети проходили возможные пути атак, и не существует других путей от источников угроз к объекту атаки, при этом из функции небезопасного состояния (ФНС) исчезнут все составляющие φ_l,

$$\varphi_l = \bigwedge_{i \in K_{\varphi l}} Z_i ,$$

которые содержали соответствующие источники угроз (т. е. можно считать, что некоторые события функции опасного состояния системы стали всегда равны нулю: $Z_k = 0$). И ФНС, соответственно, примет вид:

$$y_1(Z_1,...,Z_{m_1}) = \bigvee_{l=1}^{d_1} \varphi_l \quad \rightarrow \quad y_2(Z_1,...,Z_{m_2}) = \bigvee_{l=1}^{d_2} \varphi_l ,$$

где $m_2 < m_1$ и $d_2 < d_1$.

Но вид тех φ_l, которые не содержали $Z_k = 0$, не изменится. При этом функция вероятности кибератаки для новой ФНС, с изъятыми связями между объектами, $P\{y_2(Z_1,...,Z_{m_2}) = 1\}$, может быть получена путем замены соответствующих вероятностей $P_k = P\{Z_k = 1\}$ на $P_k = 0$ в старой ФНС $P\{y_1(Z_1,...,Z_{m_1}) = 1\}$.

Из утверждения 2 следует, что после замены $P_k = P\{Z_k = 1\}$ на $P_k = 0$ функция вероятности успешности кибератаки уменьшится (не увеличится).

3. В сценарии кибератак появляется более длинный путь атаки, без изменения источников угроз.

Это может произойти, если через изъятое ребро графа сети проходили возможные пути атак, но при этом существуют другие пути от этих источников угроз к объекту атаки, т. е. количество и последовательность действий $\{Z_k\}$, которые необходимо выполнить злоумышленнику для достижения объекта атаки, увеличится.

При этом в φ_l, которые содержали соответствующие пути атаки, увеличится количество переменных:

$$\varphi_l = \bigwedge_{i \in K_{\varphi l}} Z_i \quad \rightarrow \quad \varphi_l^* = \bigwedge_{i \in K_{\varphi l}} Z_i \wedge \{Z_k\} .$$

Это приведет к тому, что в функции вероятности успешности атаки переменная $P_k = P\{Z_k = 1\}$ заменится на произведение $P_i P_k$, и, соответственно, значение функции вероятности успешности кибератаки уменьшится (не увеличится).

Теперь продемонстрируем монотонность функции вероятности успешности кибератаки на примере.

Первоначальная структура графа G_1 сети приведена на рис. 6.7, *а* и $A = \{A, D\}$ – источник угроз; $O = \{C\}$ – объект атак.

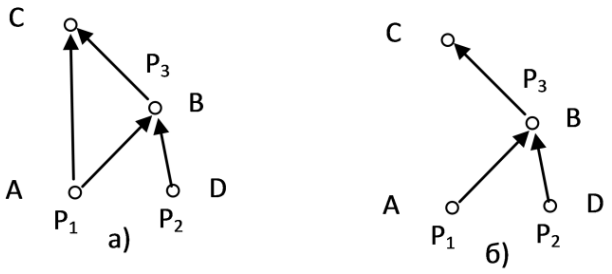


Рис. 6.7. Пути атак

Функция опасного состояния для графа G_1 будет

$$y(G_1) = Z_1 \vee Z_2 Z_3 .$$

Функция вероятности успешности кибератак для графа G_1 будет

$$P(G_1) = P_1 + (1 - P_1) P_2 P_3 .$$

Уберем ребро АС и получим граф G_2, который приведен на рис. 6.7, *б*. Таким образом, $E_2 \subset E_1$, и функция опасного состояния для графа G_2 будет

$$y(G_2) = Z_1 Z_3 \vee Z_2 Z_3 ,$$

а функция вероятности успешности кибератак для графа G_2 будет

$$P(G_2) = P_2 P_3 + (1 - P_2) P_1 P_3 .$$

Между функциями опасного состояния для графов сети G_1 и G_2 будет следующее соотношение:

$$y(G_1) \geq y(G_2) .$$

Подсчитаем разность:

$$P(G_1) - P(G_2) = P_1 + (1 - P_1) P_2 P_3 - P_2 P_3 - (1 - P_2) P_1 P_3 = P_1 - P_1 P_3 = P_1 (1 - P_3) \geq 0.$$

Итак, было показано, что функция вероятности успешности кибератаки удовлетворяет всем условиям и может быть взята в качестве целевого функционала защищенности сети.

Построение безопасной топологии сети ИКС

Таким образом, задача построения безопасной топологии сети ИКС, исходя из (6.7), будет формулироваться следующим образом:

$$\begin{cases} G^* = \arg\max\limits_{G \in O}(1 - P(G)) = \arg\min\limits_{G \in O} P(G); \\ O(G) = \{G_1, G_2, ..., G_k\} \subseteq \Omega, \end{cases} \quad (6.7)$$

где $O(G)$ – множество допустимых вариантов топологии сети;

Ω – все возможные варианты топологии сети.

В связи с отсутствием эффективного алгоритма, для определения безопасной топологии необходимо осуществить сравнение значений функции вероятности успешности кибератаки для разных вариантов графов атак, которые строятся на основе топологии сети.

Если для графов сети выполняются соотношения

$$E_1 \subset E_2 \subset ... \subset E_i \subset ...E_N,$$

то, благодаря доказанному свойству монотонности функции вероятности успешности кибератаки, будет следовать, что

$$P(G_1) \le P(G_2) \le ...P(G_i) \le ...P(G_N).$$

Соответственно, будут справедливы и следующие утверждения:

$$\forall G_1, G \in O, G_1 \subset G \Rightarrow P(G_1) \le P(G);$$

$$\forall G_2, G \in O, G_2 \subset G \Rightarrow P(G_2) \le P(G).$$

Таким образом, пользуясь утверждением 1, возможно определить лучшую топологию без непосредственного подсчета и сравнений значений функции вероятности успешности кибератаки.

Но в случае, когда один граф сети не принадлежит другому графу,

$$E_1 \not\subset E_2 \ (G_1 \not\subset G_2) \text{ и } E_2 \not\subset E_1 \ (G_2 \not\subset G_1),$$

без осуществления подсчета значений функции вероятности успешности кибератаки для данных сетей невозможно определить, какая из топологий этих сетей будет лучшей с точки зрения безопасности:

$$\left.\begin{matrix} E_1 \not\subset E_2 \\ E_2 \not\subset E_1 \end{matrix}\right| \Rightarrow P(G_1) ? P(G_2).$$

В общем случае для построения безопасной топологии сети необходимо подсчитать значения функционалов защищенности для всех графов сети в случаях, когда

$$E_i \not\subset E_j \text{ и } E_j \not\subset E_i.$$

Как было показано в (6.1) и (6.2), количество возможных вариантов топологии сети будет иметь экспоненциальный рост, в случае увеличения количества объектов в системе. При этом также необходимо учитывать, что для каждого из вариантов графа кибератак будет осуществляться построение ФНС и преобразование ее к ортогональной дизъюнктивной нормальной форме. В случае большой сети эти операции будут занимать значительное время.

Для примера, если взять сеть, содержащую N объектов, то общее количество возможных вариантов графов сети будет равно $|\Omega| = 2^{N(N-1)/2}$, без учета направлений соединения между объектами. Предположив, что вычисление значения функционала защищенности для каждого варианта графа будет занимать время *t*, тогда общее необходимое время будет иметь следующую зависимость: $T \sim t \cdot 2^{N(N-1)/2}$.

Исходя из этого, необходимо сократить количество вариантов графов сетей, для которых будет осуществляться вычисление значений функции вероятности успешности кибератаки. Сокращение вариантов может быть достигнуто за счет введения дополнительных ограничений на приемлемые варианты топологии сети. Для того чтобы определить, какие из топологий являются приемлемыми, проанализируем принципы, по которым строится топология сети ИКС.

6.4. Построение безопасной топологии ИКС на основе функции вероятности успешности кибератаки, с учетом сегментации сети

6.4.1. Модель сегментированной сети ИКС

В предыдущих пунктах, когда рассматривались топологии сети, предполагалось, что каждый из объектов сети может быть соединен любым способом с другими объектами. Но на практике реализовать такую топологию сети довольно сложно, так как это будет требовать от администратора сети сложной настройки сетевого оборудования и самих объектов.

В реальных ИКС доступ между отдельными объектами сети определяется настройками сетевого оборудования, которое управляет информационными потоками в сети (маршрутизаторы, коммутаторы, межсетевые экраны и др.), или средствами, которые реализуют эти функции программно.

Если сеть является одноранговой, то она будет полностью связной, где каждый из объектов имеет доступ ко всем другим, и вся матрица доступности объектов (рис. 6.8) будет заполнена 1.

	1	2	3	...	$N-2$	$N-1$	N
1	1	1	1	...	1	1	1
2	1	1	1	...	1	1	1
3	1	1	1	...	1	1	1
...	...	...	...	...	...	...	...
$N-2$	1	1	1	...	1	1	1
$N-1$	1	1	1	...	1	1	1
N	1	1	1	...	1	1	1

Рис. 6.8. Матрица доступности объектов ИКС

Если же сеть не является одноранговой, то она состоит из отдельных сегментов (или подсетей). В пределах каждого из этих сегментов все объекты будут иметь доступ ко всем объектам этого же сегмента. Взаимодействие и передача информации между сегментами сети будет осуществляться с помощью маршрутизаторов (или средств, которые обеспечивают маршрутизацию, – средств маршрутизации). Принципы, по которым осуществляется выделение сегментов, и технические возможности для этого рассмотрим далее.

Исходя из того, что в пределах каждого из сегментов каждый из объектов будет иметь доступ ко всем объектам этого же сегмента, при выделении отдельных сегментов в ИКС матрица доступности объектов будет принимать блочно-диагональный вид (рис. 6.9).

	v_1	v_2	v_3	...	v_{N-2}	v_{N-1}	v_N
v_1	1	1	1	...	0	0	0
v_2	1	1	1	...	0	0	0
v_3	1	1	1	...	0	0	0
...	...	...	...	...	...	...	...
v_{N-2}	0	0	0	...	1	1	1
v_{N-1}	0	0	0	...	1	1	1
v_N	0	0	0	...	1	1	1

Рис. 6.9. Матрица доступности объектов ИКС

Для общего случая, когда количество сегментов равно T, матрицу доступности объектов можно представить в такой форме (рис. 6.10).

$$\begin{pmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & & \\ 0 & & \ldots & 0 \\ 0 & & & S_T \end{pmatrix}$$

Рис. 6.10. Сегментация ИКС:

S_i – сегмент сети, который представляет собой матрицу, заполненную 1

Например, для сети, приведенной на рис. 6.11, матрица доступности будет иметь следующий вид (рис. 6.12).

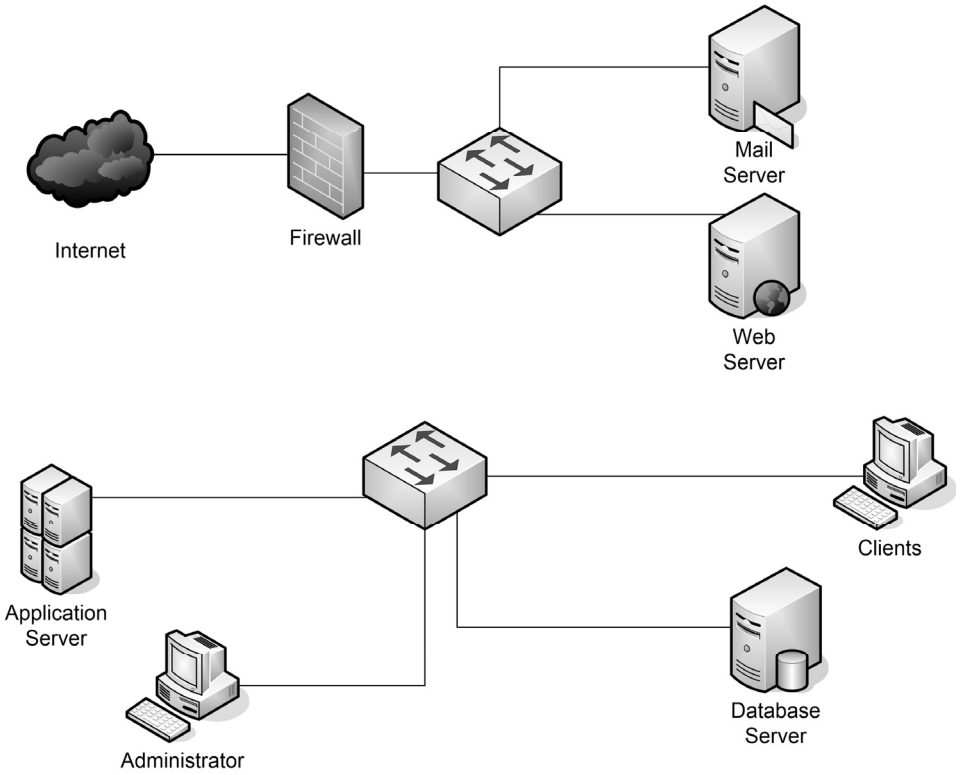


Рис. 6.11. Логическая схема сети ИКС

	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0
2	1	1	1	0	0	0	0
3	1	1	1	0	0	0	0
4	0	0	0	1	1	1	1
5	0	0	0	1	1	1	1
6	0	0	0	1	1	1	1
7	0	0	0	1	1	1	1

1 – Firewall
2 – Mail Server
3 – Web Server
4 – Administrator
5 – Clients
6 – Application Server
7 – Database Server

Рис. 6.12. Матрица доступности объектов ИКС

С точки зрения информационной безопасности, а также производительности, подобная сегментация сети была бы наилучшей. Но между отдельными объектами, которые принадлежат разным сегментам, исходя из функциональных нужд организации, должны существовать информационные потоки.

Для приведенного примера должен быть доступ от Клиентских компьютеров к Почтовому серверу и Веб-серверу, а также от Веб-сервера к серверу Баз данных (рис. 6.13)

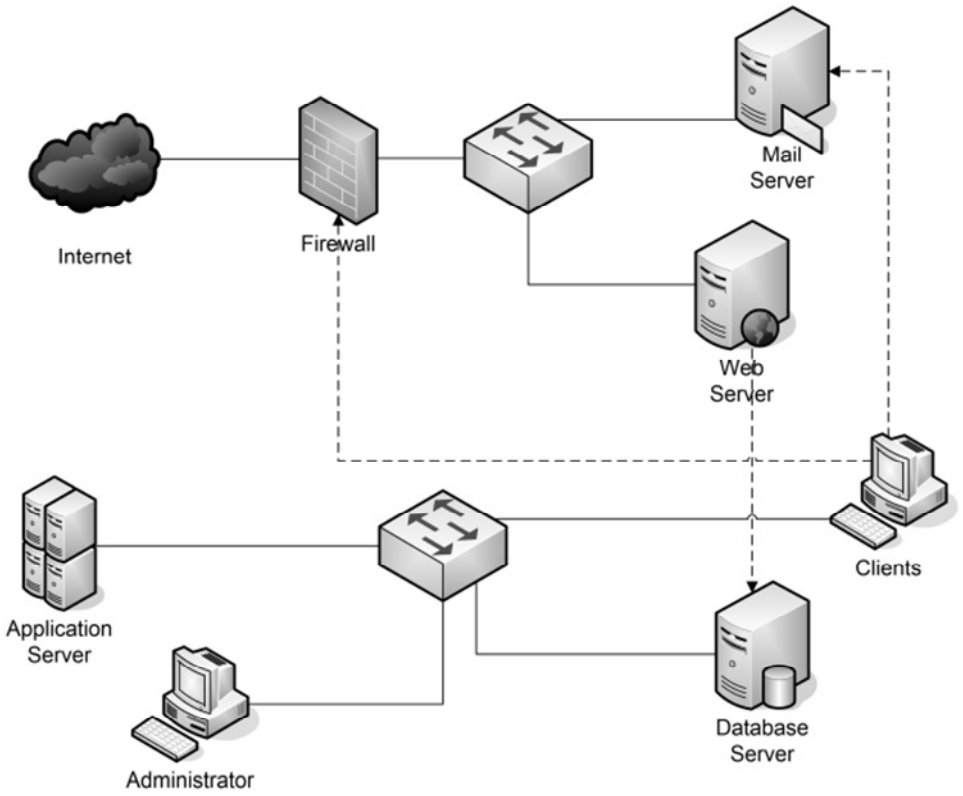


Рис. 6.13. Логическая схема сети ИКС

С учетом данных связей матрица доступности будет иметь следующий вид (рис. 6.14).

	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0
2	1	1	1	0	0	0	0
3	1	1	1	0	0	0	1
4	0	0	0	1	1	1	1
5	1	1	0	1	1	1	1
6	0	0	0	1	1	1	1
7	0	0	0	1	1	1	1

1 – Firewall
2 – Mail Server
3 – Web Server
4 – Administrator
5 – Clients
6 – Application Server
7 – Database Server

Рис. 6.14. Матрица доступности объектов ИКС

Обеспечение взаимодействия между отдельными сегментами сети (или виртуальными локальными сетями) осуществляется с использованием специальных средств маршрутизации (программных или аппаратных). Эти средства позволяют по характеристикам информационных потоков направлять их по заданному маршруту, а также по определенным признакам разрешать или нет доступ к другим объектам или сегментам сети.

Для нашего примера на маршрутизаторе необходимо прописать следующие маршруты:

– от Веб-сервера к серверу Баз данных;

– от Клиентского компьютера к Веб- и Почтовому серверу.

Но если клиентских компьютеров становится несколько и их количество планируется увеличивать, то будет целесообразным выделить их в отдельный сегмент и предоставить доступ для всех объектов этого сегмента к Серверу приложений, Почтовому серверу и Веб-серверу (рис. 6.15, 6.16).

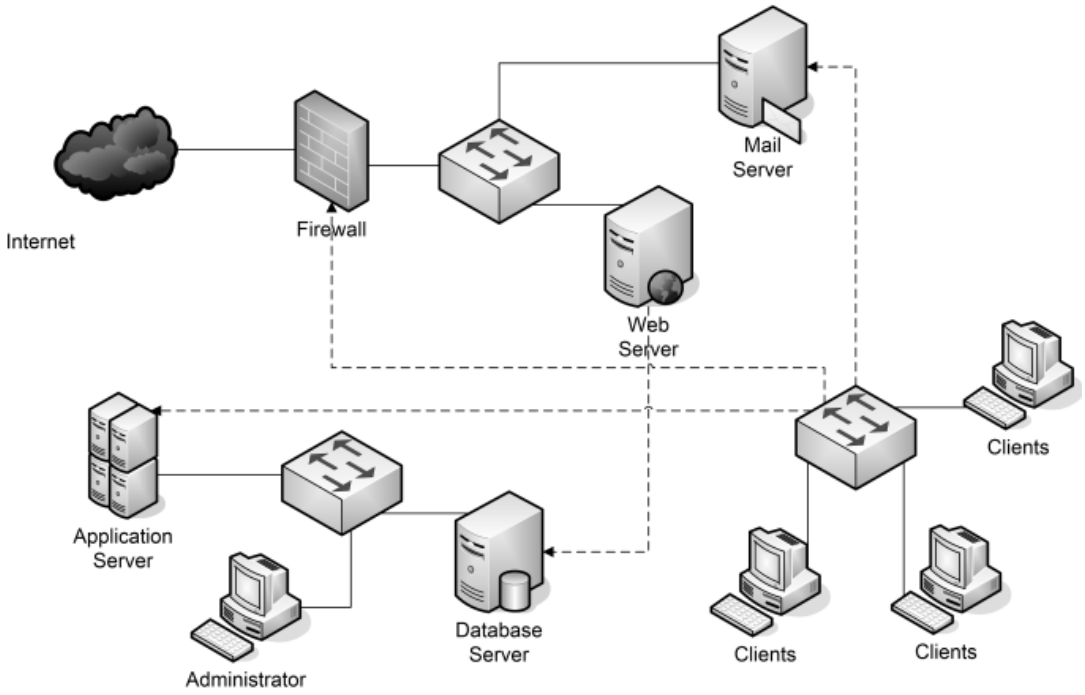


Рис. 6.15. Логическая схема сети ИКС

Таким образом, в нашей сети выделяется 3 сегмента – S_1, S_2, S_3, информационные потоки между которыми обеспечиваются с помощью маршрутизатора (рис. 6.17).

Для реальной ИКС топология сети будет состоять из отдельных сегментов $S = \{S_1, S_2, ..., S_T\}$, внутри которых каждый из объектов имеет доступ

к каждому, а правила доступа к объектам, которые находятся в других сегментах, определяются, исходя из требований к функциональности системы, и настраиваются на средствах маршрутизации.

Далее рассмотрим, каким образом в общем случае осуществляется выделение сегментов в ИКС.

	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0
2	1	1	1	0	0	0	0
3	1	1	1	0	0	0	1
4	0	0	0	1	1	1	1
5	1	1	0	0	1	1	0
6	0	0	0	1	1	1	1
7	0	0	0	1	1	1	1

1 – Firewall
2 – Mail Server
3 – Web Server
4 – Administrator
5 – Clients
6 – Application Server
7 – Database Server

Рис. 6.16. Матрица доступности ИКС

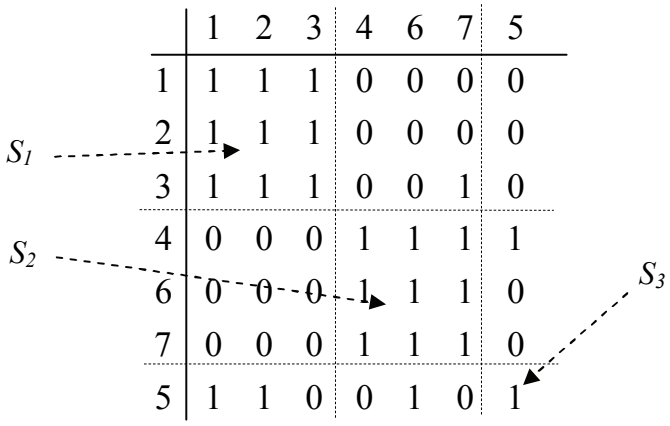


Рис. 6.17. Матрица доступности объектов ИКС с сегментами

Сегментация сети ИКС

Для разграничения доступа к отдельным ресурсам в сети эффективным средством является сегментация, которая одновременно позволяет значительно улучшить масштабируемость сети.

Существует много руководств, инструкций, практических советов от известных производителей программных и аппаратных средств по проектированию топологии сетей в зависимости от их размера, функционального назначения, территориального размещения, категорий информации и пр. Например, данный вопрос описывается в Network Security Architectures [143], документах Cisco Validated Designs [144] и Microsoft Systems Architecture [145].

Согласно рекомендациям компании Cisco [146] сегментация сети ИКС может осуществляться, исходя из следующих принципов:

1. Безопасность – выделение подсистем (объектов) с чувствительной (с точки зрения безопасности) информацией от других частей сети. Такая сегментация уменьшает вероятность несанкционированного доступа к этим данным, т. е. выделение сегментов происходит, исходя из категорий информации, которая обрабатывается в системе.

2. Проекты/программы – возможность упрощения управления проектом или работы с некоторой программой путем выделения всех его/ее пользователей в отдельный сегмент или виртуальную сеть.

3. Продуктивность/пропускная способность – объединение объектов в рамках одного сегмента сети повышает скорость обмена данными между ними.

4. Широковещательный трафик/контроль информационных потоков – так как виртуальные локальные сети не передают широковещательный трафик за свои границы, это автоматически снижает загруженность всей корпоративной сети. Также с помощью списка доступа на маршрутизаторе обеспечивается контроль и управление трафиком между сегментами сети.

5. Отделы организации – с помощью виртуальных локальных сетей можно объединять пользователей по структурным признакам организации.

Часто для сегментации сети на канальном уровне используют виртуальные локальные сети (ВЛОС, англ. VLAN). По определению, виртуальная локальная сеть – это подмножество узлов сети, трафик между которыми на канальном уровне полностью изолирован от других узлов, т. е. никакой кадр с ВЛОС не будет передаваться коммутатором за границы этой ВЛОС. ВЛОС создаются путем конфигурирования коммутаторов.

Особенностями ВЛОС является то, что трафик между ними полностью изолируется. Средствами канального уровня невозможно осуществлять контролируемый обмен информацией между разными сегментами сети. Для этих целей используются средства сетевого уровня. Передача трафика между подсетями осуществляется маршрутизаторами. При таком построении сети трафик в рамках отдельной подсети передается на канальном уровне, без привлечения средств сетевого уровня, поскольку он полностью изолирован в пределах одной ВЛОС. Коммутация на канальном уровне осуществляется быстрее, чем маршрутизация на сетевом уровне. Средства сетевого уровня

вступают в действие лишь при необходимости передавать трафик между подсетями, при этом возможно осуществлять фильтрацию трафика. Простейшая фильтрация может осуществляться непосредственно маршрутизатором. Для реализации более сложных правил фильтрации привлекаются специальные программные или программно-аппаратные средства – межсетевые экраны [4].

Таким образом, реальная ИКС будет состоять из отдельных сегментов сети, которые выделены по определенным принципам. Взаимодействие между этими сегментами осуществляется с помощью средств маршрутизации, а все объекты системы распределяются между сегментами сети, исходя из определенных соображений.

Теперь переформулируем задачу синтеза оптимальной топологии сети ИКС, учитывая сегментацию сети.

6.4.2. Построение безопасной топологии с учетом сегментации

Сначала напомним общую постановку задачи построения безопасной топологии сети. Задача построения безопасной топологии формулировалась как поиск такого варианта топологии сети, который бы обеспечивал минимум функции вероятности успешности кибератаки, причем поиск осуществлялся не среди всех возможных вариантов конфигурации сети, а лишь среди приемлемых вариантов – тех, которые обеспечивают необходимую функциональность системы и удовлетворяют дополнительным условиям.

Учитывая сегментацию, мы тем самым вводим дополнительные ограничения на конфигурацию сети и таким образом уменьшаем количество приемлемых вариантов топологии сети, для которых необходимо осуществлять вычисление значений функции вероятности успешности кибератаки:

$$\begin{cases} G^* = \arg\min\limits_{G \in O_S} P(G); \\ O_S(G) = \{G_1, G_2, ..., G_k\} \subseteq \Omega, \end{cases} \tag{6.8}$$

где $O_S(G)$ – множество допустимых вариантов топологии для сегментированной сети.

При этом если множество Ω – это всевозможные варианты топологии сети, а $O(G)$ – множество допустимых вариантов топологии сети, то будет выполняться следующее соотношение:

$$O_S(G) \subseteq O(G) \subseteq \Omega. \tag{6.9}$$

Необходимо также отметить, что оптимальные варианты топологии, полученные на основе множеств $O(G)$ и $O_S(G)$, могут отличаться, исходя из того, что $O_S(G) \subseteq O(G)$ ((6.9) и (6.8)):

$$\min_{G \in O} P(G) \le \min_{G \in O_S} P(G). \qquad (6.10)$$

Для того чтобы переформулировать задачу построения безопасной топологии сети, рассмотрим, из каких вариантов топологии будет состоять множество $O_S(G)$.

Как было показано, объекты ИКС будут распределяться между сегментами сети, т. е. сначала будет происходить выделение сегментов сети, а потом отнесение объектов системы к одному из этих сегментов. При этом такое распределение объектов между сегментами сети может быть выполнено не всегда однозначным образом: в зависимости от объекта системы, он может быть отнесен либо к одному, либо к нескольким сегментам.

Таким образом, возможные комбинации вариантов распределения объектов системы по сегментам сети и будут формировать множество допустимых вариантов топологии $O_S(G)$.

Итак, задача построения безопасной топологии сети ИКС, с учетом ее сегментации, будет формулироваться теперь следующим образом.

Распределить объекты системы по определенным сегментам сети, с учетом требований к их связанности между собой, так, чтобы обеспечить минимум функции вероятности успешности кибератаки.

Формализуем постановку данной задачи. Сначала рассмотрим процесс выделения сегментов в сети. Чаще всего выделение сегментов происходит, исходя из требований безопасности, организационной структуры предприятия и производительности сети.

Выделение сегментов, исходя из требований безопасности и организационной структуры, осуществляется преимущественно аналитическим путем, при этом также должны учитываться категории информации, которая будет обрабатываться и храниться на объектах системы. Обрабатываемые категории информации могут подпадать под определенную законодательством классификацию, а могут и не подпадать: например, открытая – присутствующая на Веб-портале; пользовательская – электронная почта, документы сотрудников; корпоративная.

Выделение сегментов, исходя из производительности сети и ее пропускной способности, может происходить как аналитически, так и с использованием формальных подходов, например, за счет измерений информационных потоков между объектами сети, как это описано в [142; 147; 148].

Учитывая то, что выделение сегментов сети выходит за рамки данной работы, будем считать, что оно уже выполнено, и мы имеем соответствующий набор $S = \{S_1, S_2, ..., S_T\}$ сегментов сети (рис. 6.18).

$$\left(\begin{array}{cc|c|c} S_1 & 0 & 0 & 0 \\ 0 & S_2 & & \\ \hline 0 & & ... & 0 \\ \hline 0 & & & S_T \end{array}\right)$$

Рис. 6.18. Матрица доступности сегментированной сети

В общем случае каждый объект системы может быть отнесен к любому из сегментов, но в реальной ситуации возможность объекта принадлежать какому-нибудь из сегментов будет определяться его назначением, территориальным нахождением, проектами и информацией, которая будет на нем обрабатываться, а также другими факторами.

Объект v_j, исходя из его свойств, может быть отнесенным к некоторому подмножеству сегментов $\{S_{j_1}, S_{j_2}, ..., S_{j_s}\}$.

Это будет распространяться не на все объекты системы $V = \{v_1, ..., v_N\}$.

Таким образом, будет определена таблица $(V \mid S)$ (табл. 6.1) возможной принадлежности объектов $V = \{v_1, ..., v_N\}$ сегментам сети $S = \{S_1, S_2, ..., S_T\}$.

Таблица 6.1

Распределение объектов по сегментам сети

	S_1	S_2	...	S_T
v_1	{1, 0}	{1, 0}	...	{1, 0}
v_2	{1, 0}	{1, 0}	...	{1, 0}
...	...	...	...	...
v_N	{1, 0}	{1, 0}	...	{1, 0}

По вертикали табл. 6.1 будут находиться сегменты сети, а по горизонтали – объекты сети. Значение на пересечении строк и столбцов может быть 0 или 1 и будет определяться функцией

$$f_{ji} = f(v_j \mid S_i) = \begin{cases} 1, \text{если объект } v_j \text{ можно отнести к сегменту } S_i \\ 0, \text{если объект } v_j \text{ нельзя отнести к сегменту } S_i \end{cases}; j = \overline{1,N}, i = \overline{1,T}.$$

Строку данной таблицы, где функция f_{ji} будет равна 1 ($f_{ji} = f(v_j \mid S_i) = 1$), обозначим как множество $I_j = \{(v_j \mid S_{j_1}), (v_j \mid S_{j_2}), ..., (v_j \mid S_{j_z})\} = \{v_j \mid S_{j_1}, S_{j_2}, ..., S_{j_z} \in S\}$, в которое входят все сегменты $\{S_{j_1}, S_{j_2}, ..., S_{j_z}\} \subseteq S$, к которым может принадлежать объект v_j.

Или если записать в более общем виде:

$$I_j = \{(v_j \mid S_i) \mid f_{ji} = 1, i = \overline{1,T}\} = \{(v_j \mid S_{j_1}), (v_j \mid S_{j_2}), ..., (v_j \mid S_{j_z})\} =$$
$$= \{v_j \mid S_{j_1}, S_{j_2}, ..., S_{j_z} \in S\},$$

где $j = \overline{1,N}$.

Элементы множества I_j будем обозначать как пару: объект – сегмент сети ($s_{jk} = (v_j \mid S_k)$), для которого значения $f_{ji} = f(v_j \mid S_i) = 1$:

$$I_j = \{s_{ji} = (v_j \mid S_i) \mid f_{ji} = 1, i = \overline{1,T}\} = \{(v_j \mid S_{j_1}), (v_j \mid S_{j_2}), ..., (v_j \mid S_{j_z})\} =$$
$$= \{s_{j1}, s_{j2}, ..., s_{jz}\} \quad (6.11)$$

Тогда каждый из вариантов распределения объектов по сегментам сети будет определять отдельную топологию G_i:

$$G_i = (s_{jk} = (v_j \mid S_k) \mid s_{jk} \in I_j, j = \overline{1,N}), \quad (6.12)$$

где $G_i \in O_S(G)$, $i = \overline{1, |O_S|}$.

6.5. Процедура построения безопасной топологии сети

Приведем процедуру построения безопасной топологии с учетом сегментации сети:

1. Входными данными являются следующие наборы:

– объекты сети – $V = \{v_1, ..., v_N\}$;

– множество источников угроз – $A = (a_1,...,a_K) \subset V$ и объектов кибератак – $O = (o_1,...,o_M) \subset V$;

– вероятности событий Z_i захвата объектов v_i – $P(Z_i = 1) = P_i$;

– требования функциональной связанности между объектами.

2. Определение обязательных связей между объектами, исходя из требований функциональной связанности между ними – $\{(v_a, v_b)\}$.

3. Выделение сегментов сети – $S = \{S_1, S_2,..., S_T\}$.

4. Для каждого объекта v_j, учитывая информацию, которая будет храниться или обрабатываться в пределах объекта, записать таблицу (табл. 6.1) распределения объектов по сегментам $(V \mid S)$ и для каждого объекта сформировать множества I_j (6.11), где

$$I_j = \left\{ s_{ji} = \left(v_j \mid S_i\right) \mid f_{ji} = 1, i = \overline{1,T} \right\} = \left\{ (v_j \mid S_{j_1}), (v_j \mid S_{j_2}),..., (v_j \mid S_{j_z}) \right\} = \left\{ s_{j1}, s_{j2},..., s_{jz} \right\}.$$

5. Сформировать множество вариантов топологии $O_S(G)$ (6.12), где каждый из вариантов топологии G_i будет иметь вид $G_i = \left(s_{jk} = (v_j \mid S_k) \mid s_{jk} \in I_j, j = \overline{1,N} \right)$.

6. Для каждого варианта топологии G_i записать функцию вероятности успешности кибератаки $P(G_i)$.

7. Путем подсчета значений функций вероятностей успешности кибератаки выбрать безопасную топологию (или их набор) (6.8):

$$\begin{cases} G^* = \arg\min\limits_{G \in O_S} P(G); \\ O_S(G) = \{G_1, G_2,..., G_k\} \subseteq \Omega. \end{cases}$$

Для оценки времени работы данного алгоритма подсчитаем количество вариантов топологии сети, которые содержатся в множестве $O_S(G) = \{G_1, G_2,..., G_k\}$, а также их возможное максимальное количество.

Каждый из вариантов топологии G_i будет представлять собой один из вариантов размещения объектов V по сегментам сети L. Общее количество вариантов будет равно произведению количества сегментов, к которым может быть отнесен каждый из N объектов:

$$\left| O_S(G) \right| = \prod_{j=1}^{N} \left| I_j \right|.$$

Если предположить, что каждый из объектов может принадлежать к каждому из T сегментов сети, то максимальное значение возможных вариантов топологии будет равняться T^N. Из этого будет вытекать следующее соотношение для количества возможных вариантов топологии сети:

$$\left|O_S\right| = \prod_{j=1}^{N} \left|I_j\right| \le T^N,$$

где N – количество объектов сети;

$\left|I_j\right|$ – количество сегментов, к которым может принадлежать объект v_j;

T – общее количество сегментов сети.

В среднем большинство объектов могут быть размещены в трех сегментах, тогда количество вариантов будет равно $\left|O_S\right| \le 3^N$.

Напомним, что максимальное количество вариантов, среди которых необходимо выбрать оптимальную топологию, вначале было равно $2^{N(N-1)/2}$. Таким образом, применяя сегментацию сети, нам удалось эту зависимость сократить до 3^N. Хотя вид зависимости остался экспоненциальным, теперь стало возможным применение данного алгоритма для сетей среднего размера. Также нужно учитывать то, что только некоторые из объектов могут быть отнесены к нескольким сегментам. Преобладающее количество объектов будет относиться лишь к одному из сегментов. Кроме этого, однотипные объекты системы (например, клиентские рабочие места одного подразделения) можно группировать и считать как один объект.

6.6. Пример построения безопасной топологии сети и анализ полученных результатов

Для демонстрации работы алгоритма рассмотрим следующий пример. Возьмем систему, которая содержит семь объектов – $V = \{v_1,...,v_7\}$. Также определим:

– множество источников угроз – $A = \{v_1, v_5\}$;

– множество объектов кибератаки – $O = \{v_3, v_7\}$;

– множество обязательных связей между объектами системы $\{(v_a, v_b)\}$ зададим следующей матрицей (рис. 6.19).

	1	2	3	4	5	6	7
1	1	1	1	–	–	–	–
2	1	1	–	–	–	–	–
3	1	–	1	–	–	–	1
4	–	–	–	1	–	1	–
5	1	1	–	–	1	1	–
6	–	–	–	1	–	1	1
7	–	–	–	–	–	–	1

1 – Firewall
2 – Mail Server
3 – Web Server
4 – Administrator
5 – Clients
6 – Application Server
7 – Database Server

Рис. 6.19. Матрица доступности с обязательными связями между объектами:
«1» – наличие обязательной связи между соответствующими объектами;
«–» – возможность наличия или отсутствия связи

Выделим в данной сети три сегмента $S = \{S_1, S_2, S_3\}$. Соответственно, матрица доступности объектов будет состоять из трех блоков:

$$\begin{pmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{pmatrix}.$$

Сегментация сети была сделана, исходя из того, какая информация будет преимущественно обрабатываться в каждом сегменте. Соответственно, объект может быть отнесен к данному сегменту, если на нем будет обрабатываться или храниться соответствующий тип информации. Если же на объекте будут обрабатываться или храниться несколько типов информации, то этот объект может быть отнесен к нескольким соответствующим сегментам сети.

Возможные распределения объектов по сегментам сети приведены в табл. 6.2.

Таблица 6.2

Распределение объектов по сегментам сети

	Открытая информация S_1	**Информация пользователей S_2**	**Корпоративная информация S_3**
Firewall – v_1	1	0	0
Mail Server – v_2	1	1	0
Web Server – v_3	1	0	0
Administrator – v_4	0	0	1
Clients – v_5	1	1	1
Application Server – v_6	0	0	1
Database Server – v_7	1	0	1

Таким образом, построена таблица распределения $(V \mid S)$. Исходя из табл. 6.2, запишем значения множества I_j:

$I_1 = (v_1 \mid S_1)$;

$I_2 = (v_2 \mid S_1, S_2)$;

$I_3 = (v_3 \mid S_1)$;

$I_4 = (v_4 \mid S_3)$;

$I_5 = (v_5 \mid S_1, S_2, S_3)$;

$I_6 = (v_6 \mid S_3)$;

$I_7 = (v_7 \mid S_1, S_3)$.

Если бы каждый из объектов мог принадлежать каждому из сегментов (без учета значений табл. 6.2), общее количество возможных вариантов топологии было бы

$$\left|O\right| = T^N = 3^7 = 2187.$$

С учетом ограничений, которые накладывает табл. 6.2, количество вариантов топологии сети будет

$$\left|\Omega\right| = \prod_{j=1}^{N} \left|I_j\right| = 1*2*1*1*3*1*2 = 12.$$

Эти возможные 12 вариантов топологии можно также представить в виде путей на графе (рис. 6.20), каждый из этих путей представляет собой один из вариантов распределения объектов по сегментам сети.

Основываясь на значениях множества I_j (и построенном графе) и множества обязательных связей между объектами $\{(v_a, v_b)\}$, построим множество $O_S(G)$ и возможные варианты топологии сети G_i:

$G_1 = \{(v_1 \mid S_1), (v_2 \mid S_1), (v_3 \mid S_1), (v_4 \mid S_3), (v_5 \mid S_1), (v_6 \mid S_3), (v_7 \mid S_1)\}$;

$G_2 = \{(v_1 \mid S_1), (v_2 \mid S_2), (v_3 \mid S_1), (v_4 \mid S_3), (v_5 \mid S_1), (v_6 \mid S_3), (v_7 \mid S_1)\}$;

$G_3 = \{(v_1 \mid S_1), (v_2 \mid S_1), (v_3 \mid S_1), (v_4 \mid S_3), (v_5 \mid S_2), (v_6 \mid S_3), (v_7 \mid S_1)\}$;

$G_4 = \{(v_1 \mid S_1), (v_2 \mid S_1), (v_3 \mid S_1), (v_4 \mid S_3), (v_5 \mid S_3), (v_6 \mid S_3), (v_7 \mid S_1)\}$;

$G_5 = \{(v_1 \mid S_1), (v_2 \mid S_1), (v_3 \mid S_1), (v_4 \mid S_3), (v_5 \mid S_1), (v_6 \mid S_3), (v_7 \mid S_3)\}$;

$G_6 = \{(v_1 \mid S_1), (v_2 \mid S_2), (v_3 \mid S_1), (v_4 \mid S_3), (v_5 \mid S_2), (v_6 \mid S_3), (v_7 \mid S_1)\}$;

$$G_7 = \{(v_1 \mid S_1),(v_2 \mid S_2),(v_3 \mid S_1),(v_4 \mid S_3),(v_5 \mid S_2),(v_6 \mid S_3),(v_7 \mid S_3)\};$$

$$G_8 = \{(v_1 \mid S_1),(v_2 \mid S_2),(v_3 \mid S_1),(v_4 \mid S_3),(v_5 \mid S_3),(v_6 \mid S_3),(v_7 \mid S_1)\};$$

$$G_9 = \{(v_1 \mid S_1),(v_2 \mid S_2),(v_3 \mid S_1),(v_4 \mid S_3),(v_5 \mid S_3),(v_6 \mid S_3),(v_7 \mid S_3)\};$$

$$G_{10} = \{(v_1 \mid S_1),(v_2 \mid S_1),(v_3 \mid S_1),(v_4 \mid S_3),(v_5 \mid S_3),(v_6 \mid S_3),(v_7 \mid S_3)\};$$

$$G_{11} = \{(v_1 \mid S_1),(v_2 \mid S_2),(v_3 \mid S_1),(v_4 \mid S_3),(v_5 \mid S_1),(v_6 \mid S_3),(v_7 \mid S_3)\};$$

$$G_{12} = \{(v_1 \mid S_1),(v_2 \mid S_1),(v_3 \mid S_1),(v_4 \mid S_3),(v_5 \mid S_2),(v_6 \mid S_3),(v_7 \mid S_3)\};$$

$$O_S(G) = \{G_1, G_2, ..., G_{12}\}.$$

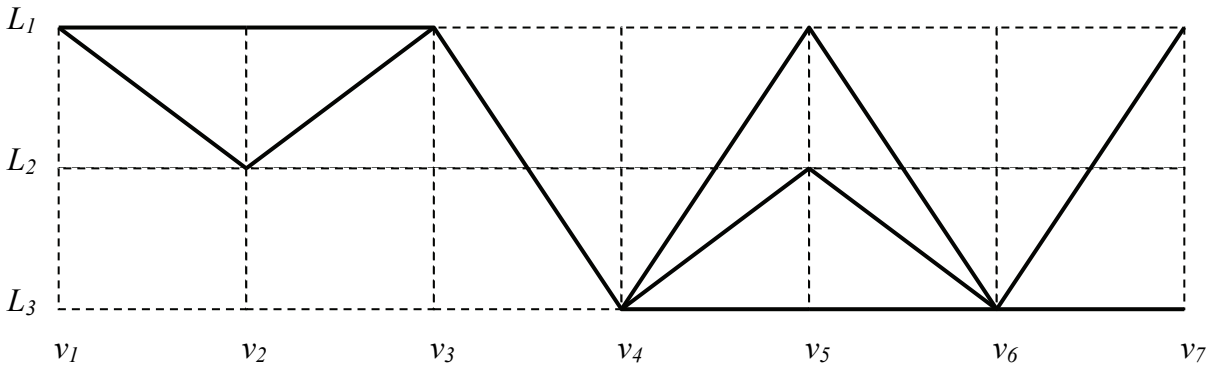


Рис. 6.20. Возможные комбинации распределения объектов по сегментам сети

Таким образом, заданы все необходимые входные данные для построения множества функций вероятности успешности кибератаки для каждой из топологий G_i:

$$P(G_1) = P_1P_3 + (1-P_1)P_3P_5 + P_1(1-P_3)P_7 + (1-P_1)(1-P_3)P_5P_7;$$

$$\ldots$$

$$P(G_7) = P_1P_3 + (1-P_1)P_5P_6P_7 + P_1(1-P_3)P_5P_6P_7;$$

$$\ldots$$

$$P(G_{12}) = P_1P_3 + (1-P_1)P_2P_3P_5 + (1-P_1)(1-P_2)P_5P_6P_7 + (1-P_1)P_2(1-P_3)P_5P_6P_7 +$$

$$+ P_1(1-P_2)(1-P_3)P_5P_6P_7 + P_1P_2(1-P_3)P_5P_6P_7.$$

Все 12 вариантов функций вероятности успешности кибератаки приводить не будем, а сразу подсчитаем их значения. Для этого в табл. 6.3 определим два варианта значений вероятностей захвата объектов системы: $P1$ и $P2$. В первой строке таблицы все вероятности захвата являются равными 0,5, а во второй – приближенными к реальным значениям.

Таблица 6.3

Вероятности захвата объектов

	P_1	P_2	P_3	P_4	P_5	P_6	P_7
$P1$	0,5	0,5	0,5	0,5	0,5	0,5	0,5
$P2$	0,3	0,5	0,7	0,1	0,9	0,5	0,3

Основываясь на этих данных, получим следующие 12 значений целевого функционала защищенности в зависимости от топологий (табл. 6.4).

Таблица 6.4

Значения функции вероятности успешности кибератак для разных топологий

	$P(G_i)$ для $P1$	$P(G_i)$ для $P2$
G_1	0,5625	0,7347
G_2	0,5625	0,7347
G_3	0,5	0,5331
G_4	0,5	0,5331
G_5	0,4375	0,6915
G_6	0,4375	0,3315
G_7	0,34375	0,31665
G_8	0,4375	0,3315
G_9	0,4375	0,4233
G_{10}	0,46875	0,57765
G_{11}	0,4375	0,6915
G_{12}	0,390625	0,504075

Построим также соответствующую диаграмму (рис. 6.21).

Проанализируем полученные результаты. Во-первых, исходя из табл. 6.4 и рис. 6.21, отметим, что для разных входных значений вероятностей захвата объектов (ряды $P1$ и $P2$) соотношения между результирующими значениями целевых функционалов защищенности могут существенно отличаться, хотя и наблюдается определенная корреляция. При одинаковой топологии значения функции вероятности успешности кибератаки являются большими для ряда $P1$ (G_6, G_7, G_8), а в некоторых случаях для – $P2$ (G_1, G_2, G_5).

Из этого следует, что для синтеза оптимальной топологии значения вероятностей захвата объектов нужно задавать близкие к реальным.

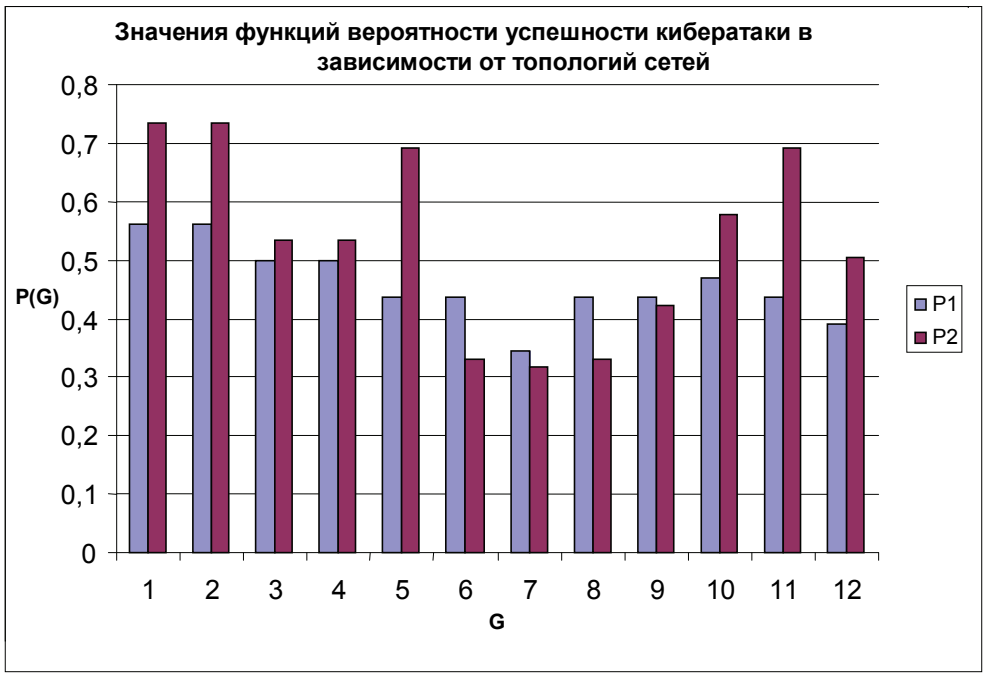


Рис. 6.21. Значения функций вероятности успешности кибератаки в зависимости от топологий сетей

Во-вторых, в нашем примере, в независимости от вероятностей захвата объектов, минимум функции вероятности успешности кибератаки будет достигаться при топологии G_7, а максимум – при G_1, G_2.

При топологиях G_1, G_2 большинство объектов будут сосредоточены в сегменте S_1, среди этих объектов будут как источники угроз, так и объекты кибератак, поэтому данные топологии будут наихудшими с точки зрения безопасности. Матрицы доступности для них будут следующие (рис. 6.22).

	1	2	3	5	7	4	6
1	1	1	1	1	1	0	0
2	1	1	1	1	1	0	0
3	1	1	1	1	1	0	0
5	1	1	1	1	1	0	1
7	1	1	1	1	1	0	0
4	0	0	0	0	0	1	1
6	0	0	0	0	1	1	1

G_1

	1	3	5	7	2	4	6
1	1	1	1	1	1	0	0
3	1	1	1	1	0	0	0
5	1	1	1	1	1	0	1
7	1	1	1	1	0	0	0
2	1	0	0	0	1	0	0
4	0	0	0	0	0	1	1
6	0	0	0	1	0	1	1

G_2

Рис. 6.22. Матрицы доступности объектов

Оптимальной топологией с точки зрения безопасности будет $G^* = \arg\min_{G \in O_S} P(G) = G_7$. Матрица доступности для данной топологии будет иметь следующий вид (рис. 6.23).

	1	3	2	5	4	6	7
1	1	1	1	0	0	0	0
3	1	1	0	0	0	0	1
2	1	0	1	1	0	0	0
5	1	0	1	1	0	1	0
4	0	0	0	0	1	1	1
6	0	0	0	0	1	1	1
7	0	0	0	0	1	1	1

G_7

1 – Firewall
2 – Mail Server
3 – Web Server
4 – Administrator
5 – Clients
6 – Application Server
7 – Database Server

Рис. 6.23. Матрица доступности объектов

При данной топологии объекты будут распределены по сегментам сети следующим образом:

$S_1 = \{v_1, v_3\}$ = {Firewall, Web Server};

$S_2 = \{v_2, v_5\}$ = {Mail Server, Clients};

$S_3 = \{v_4, v_6, v_7\}$ = {Administrator, Application Server, Database Server}.

Исходя из рекомендаций по построению топологии сети и практических соображений, странным является то, что объект v_2 (Mail Server) оказался не в сегменте S_1, а в сегменте S_2 вместе с клиентскими рабочими станциями. Обычно на практике почтовый сервер размещается в демилитаризованном сегменте и иногда даже совмещается с Веб-сервером. Но такое совмещение нежелательно, так как, захватив один из серверов, атакующий сразу же получит доступ и к другому серверу. Таким образом, лучше, чтобы эти серверы были отделены один от другого.

Теперь посмотрим на топологию G_{12}. Здесь предлагается вариант размещения серверов, который чаще всего рекомендуют использовать на практике, а именно:

$S_1 = \{v_1, v_2, v_3\}$ = {Firewall, Mail Server, Web Server};

$S_2 = \{v_5\}$ = {Clients};

$S_3 = \{v_4, v_6, v_7\}$ = {Administrator, Application Server, Database Server}.

Но, исходя из полученных результатов, этот вариант топологии не будет оптимальным. Если в данном варианте размещения объектов по сегментам сети в сегменте S_1 убрать связь между Веб-сервером и Почтовым сервером, как это показано в матрице доступности (рис. 6.24), то функция вероятности успешности атаки будет принимать следующие значения:

$$P(G_{12}^*)\big|_{P1} = 0{,}34375;$$

$$P(G_{12}^*)\big|_{P2} = 0{,}31665.$$

	1	3	2	5	4	6	7
1	1	1	1	0	0	0	0
3	1	1	0	0	0	0	1
2	1	0	1	0	0	0	0
5	1	0	1	1	0	1	0
4	0	0	0	0	1	1	1
6	0	0	0	0	1	1	1
7	0	0	0	0	1	1	1

G_7

1 – Firewall
2 – Mail Server
3 – Web Server
4 – Administrator
5 – Clients
6 – Application Server
7 – Database Server

Рис. 6.24. Матрицы доступности объектов

Видим, что данные значения будут равны оптимальным значениям, полученным для топологии G_7. Это свидетельствует о том, что путем совмещения формального подхода и устоявшихся практических решений можно получать новые результаты.

Таким образом, разработанную модель и процедуру можно использовать на практике как вспомогательное аналитическое средство во время проектирования топологии ИКС.

Выводы

В данной главе рассматривается построение безопасной топологии ИКС как поиск такого графа сети, при котором функция вероятности успешности кибератаки принимает минимальное значение при ограничениях на приемлемые варианты топологии и фиксированных значениях вероятностей захвата объектов.

На тестовом примере показано, что предложенная модель и процедура в результате применения дают топологию сети, которая совпадает с той, что предлагается в руководствах по проектированию безопасных сетей.

7. ОПТИМАЛЬНОЕ РАЗМЕЩЕНИЕ МЕХАНИЗМОВ ЗАЩИТЫ В УЗЛАХ ИКС

В данной главе решается задача оптимального размещения механизмов защиты в ИКС при ограничении на стоимость кибернетической системы защиты информации, где в качестве целевой функции используется модифицированная функция вероятности успешности кибератаки.

Для решения задачи оптимального размещения механизмов защиты применяется алгоритм последовательного анализа вариантов.

7.1. Механизмы защиты и функция вероятности успешности кибератаки

Будем использовать функцию вероятности успешности кибератаки в качестве целевого функционала для решения задачи оптимального размещения механизмов защиты в узлах ИКС. Также рассмотрим, как механизмы защиты и их характеристики могут быть учтены в данной функции. Для этого вспомним, как была построена функция вероятности успешности кибератаки.

Сначала было определено понятие опасного состояния, в которое перейдет ИКС в случае успешной кибератаки (последовательности действий, которые должен выполнить злоумышленник), если хотя бы один из ее важных компонентов (объектов) перейдет в опасное состояние (5.11). Потом была построена ФОС (5.15) в виде функции алгебры логики. Дальше ФОС приводилась к ОДНФ и непосредственной заменой булевых переменных на их вероятностные значения – к вероятностному виду (5.17).

Рассмотрим более детально переход от булевых переменных к их вероятностным значениям – вероятностям захвата объектов ИКС (5.7) – (5.8).

В терминах булевых переменных событие захвата объекта трактовалось в бинарном виде: состоялось или не состоялось, т. е. захвачен объект, расположенный на пути кибератак, или не захвачен. При переходе к вероятностной форме для каждого из событий записывалась вероятность того, что данное событие состоится, т. е. оценивалась вероятность захвата объекта.

При использовании механизмов защиты данные механизмы будут влиять на значение вероятности (в сторону ее уменьшения) захвата объекта.

Согласно п. 5.3 и соотношению (5.10), с учетом механизмов защиты вероятность захвата объекта $v_i \in V$ будет определяться как

$$P_i^{'} = P_i(\overline{M_i K_i}) = P_i(1 - M_i K_i),$$

где $P_i \in [0,1]$ – вероятность захвата объекта;

$M_i = \{0,1\}$ – наличие или отсутствие механизма защиты;

$K_i \in [0,1]$ – коэффициент эффективности механизма защиты (вероятность того, что механизм защиты M_i сможет противодействовать кибератакам, направленным на захват объекта v_i).

Используя механизм защиты M_i, можно свести вероятность захвата P_i объекта v_i полностью к нулю или снизить в несколько раз. Эффективность механизмов защиты будет зависеть от ряда характеристик, индивидуальных для каждого из типов механизмов защиты.

На основе графа кибератак (5.6) систему защиты, используя модель системы защиты с полным перекрытием [88], можно представить в виде кортежа, который назовем графом киберзащиты:

$$G(V, E, A, O, P, M, K). \qquad (7.1)$$

Если захват объекта системы может произойти несколькими путями, то необходимо использовать несколько разных механизмов защиты. Такая ситуация, с возможностью выбора между механизмами защиты, может быть представлена с помощью графа киберзащиты следующим образом (рис. 7.1).

На рис. 7.1 события Z_1, Z_2, Z_3 означают захват соответствующих объектов системы (v_1, v_2, v_3), при этом захват объекта v_1 может произойти двумя разными путями: за реализацию первого из них отвечает событие Z_{11}, второго – Z_{12}. Тогда реализация события Z_1 может быть представлена как $Z_1 = Z_{11} \vee Z_{12}$.

Для защиты от захвата объектов для событий Z_{11} и Z_3 могут применяться разные механизмы защиты, которые имеют разные характеристики коэффициентов эффективности. Для Z_{11} это будут $K_{11-1}, K_{11-2}, K_{11-3}$, а для Z_3 – $K_{3-1}, K_{3-2}, K_{3-3}, K_{3-4}$.

Но в конечном случае вероятность каждого из событий может быть представлена в виде (5.10), где окончательное значение вероятности захвата

будет зависеть от коэффициента эффективности механизма защиты и начальной вероятности захвата.

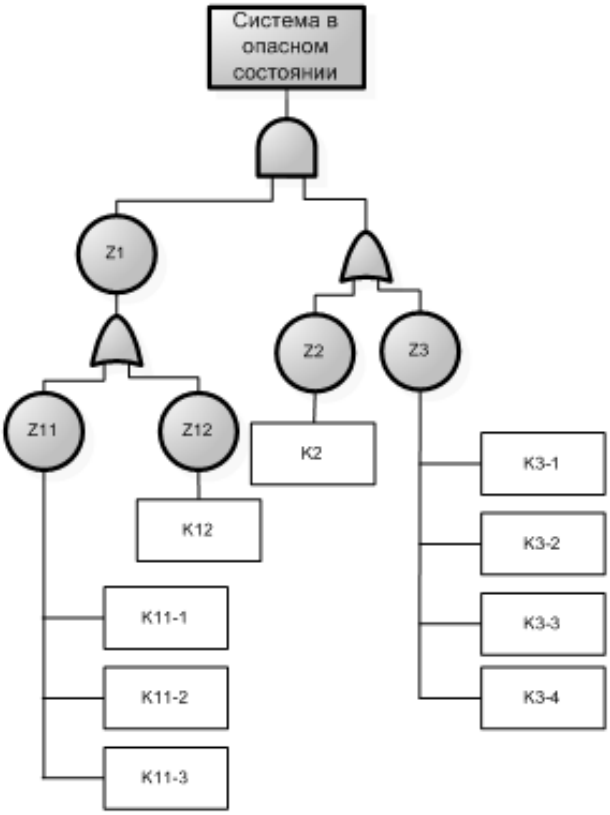


Рис. 7.1. Граф киберзащиты на основе графа кибератак

Для приведенного на рис. 7.2 примера функция опасного состояния будет определяться следующим выражением:

$$y = (Z_{11} \vee Z_{12})(Z_2 \vee Z_3)\,. \tag{7.2}$$

На основе (7.2) функция вероятности успешности кибератаки будет иметь вид

$$\begin{aligned} P = {} & P_{12}P_3 + P_{11}(1-P_{12})P_3 + (1-P_{11})P_{12}(1-P_3)P_2 + P_{11}P_{12}P_2(1-P_3) + \\ & + P_{11}(1-P_{12})P_2(1-P_3)\,. \end{aligned} \tag{7.3}$$

После применения механизмов защиты с соответствующими коэффициентами эффективности функция вероятности успешности кибератаки (7.3), согласно (5.10), примет следующий вид:

$$\begin{aligned} & P(P,M,K) = \\ & = P_{12}(1-M_{12}K_{12})P_3(1-M_3K_3) + \\ & + P_{11}(1-M_{11}K_{11})(1-P_{12}(1-M_{12}K_{12}))P_3(1-M_3K_3) + \\ & + (1-P_{11}(1-M_{11}K_{11}))P_{12}(1-M_{12}K_{12})(1-P_3(1-M_3K_3))P_2(1-M_2K_2) + \\ & + P_{11}(1-M_{11}K_{11})P_{12}(1-M_{12}K_{12})P_2(1-M_2K_2)(1-P_3(1-M_3K_3)) + \\ & + P_{11}(1-M_{11}K_{11})(1-P_{12}(1-M_{12}K_{12}))P_2(1-M_2K_2)(1-P_3(1-M_3K_3))\,. \end{aligned} \tag{7.4}$$

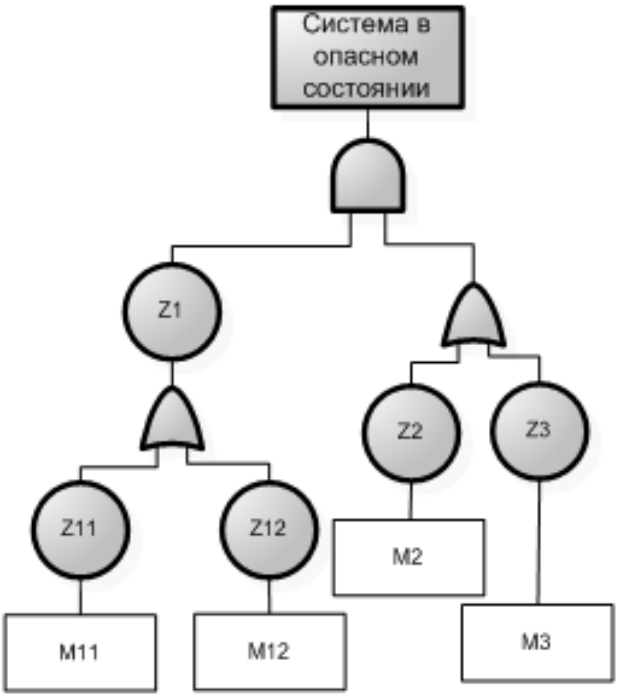


Рис. 7.2. Граф киберзащиты

Таким образом, функцию вероятности успешности кибератаки, с учетом механизмов защиты, можно получить на основе (5.17) путем подстановки в нее формулы (5.10) вместо вероятностей захвата объектов:

$$P(G,P,M,K). \qquad (7.5)$$

В соотношении (7.5) функции вероятности успешности кибератаки учитываются механизмы защиты, которые используются в ИКС, и их коэффициенты эффективности.

Исходя из утверждения 2, которое было доказано в п. 5.7, модифицированная функция вероятности (7.5) будет возрастать при увеличении значений вероятностей P_i и ниспадать при увеличении коэффициентов эффективности K_i.

Если все коэффициенты эффективности будут равны 0, то функция (7.5) будет совпадать с функцией вероятности успешности кибератак (5.17), без учета механизмов защиты:

$$P(G,P,M,K)\big|_{K=0} = P(G,P).$$

При всех коэффициентах эффективности, равных 1, значение функции (7.5) будет равно 0:

$$P(G,P,M,K)\big|_{K=1} = 0.$$

Такой случай возможен в теории, но в реальности не может быть реализован.

Таким образом, функция вероятности успешности кибератаки $P(G,P,M,K)$ будет содержать переменные, отвечающие за механизмы защиты, при этом для защиты каждого из объектов может иметься выбор среди нескольких механизмов защиты, которые будут отличаться коэффициентами эффективности и стоимостью.

При отсутствии ограничений на стоимость системы киберзащиты можно было бы применить все механизмы защиты ($M_i = 1$), причем с максимальными коэффициентами эффективности. Тогда значение функции вероятности успешности кибератаки стало бы минимальным:

$$P_{\min}(G,P,K) = P(G,P,M,K)\big|_{M=1}.$$

Если же не применять ни одного из механизмов защиты ($M_i = 0$), то стоимость системы киберзащиты будет равняться 0. В этом случае значение функции вероятности успешности кибератаки будет зависеть лишь от вероятностей захвата объектов и будет совпадать с функцией вероятности успешности кибератак (5.17), и будет максимальным:

$$P_{\max}(G,P) = P(G,P,M,K)\big|_{M_i=0}.$$

Таким образом, будет справедливо соотношение

$$0 = P(G,P,M,K)\big|_{K,M=1} \le P(G,P,K) = P(G,P,M,K)\big|_{M=1} \le P(G,P) =$$
$$= P(G,P,M,K)\big|_{K,M=0}$$

и, соответственно,

$$P_{\min}(G,P,K) \le P(G,P,M,K) \le P_{\max}(G,P).$$

Исходя из приведенных выше свойств, модифицированная функция вероятности успешности кибератаки $P(G,P,M,K)$ может быть использована в качестве целевого функционала при решении задачи оптимального размещения механизмов защиты в ИКС. Приведем постановку данной задачи в следующем пункте.

7.2. Постановка задачи оптимального размещения механизмов защиты при ограничении на их стоимость

Задачу оптимального размещения механизмов защиты при ограничении на их стоимость сформулируем как оптимальный выбор механизмов защиты в ИКС, с учетом их коэффициентов эффективности, при известных сценариях кибератак и ограничении на совокупную стоимость системы киберзащиты. Данная задача является одной из задач синтеза систем киберзащиты информации с оптимальными свойствами [99].

В общем виде постановка данной задачи заключается в следующем:

$$\begin{cases} F(M,K) \to \max\limits_{M}; \\ \sum\limits_{i} C_{M_i} \le C_{\max}, \end{cases} \tag{7.6}$$

где $F(M,K)$ – целевой функционал, который будет характеризовать защищенность системы и будет зависеть от выбранных механизмов защиты M, их размещения в ИКС, а также коэффициентов эффективности K;

C_{M_i} – стоимость выбранного механизма защиты;

$C_{\max}$ – максимальная совокупная стоимость системы киберзащиты.

В качестве целевого функционала возьмем функцию вероятности успешности кибератаки (7.5), при этом будем рассматривать задачу по выбору оптимальных механизмов защиты при фиксированной топологии сети – $P(P,M,K) = P(G^*,P,M,K)$.

Тогда задача оптимального размещения механизмов защиты будет состоять в выборе лишь тех механизмов защиты, которые при ограничении на их совокупную стоимость будут минимизировать значение функции вероятности успешности кибератаки (7.5). На основе соотношения (7.6) данная задача будет формулироваться следующим образом:

$$\begin{cases} P(P,M,K) \to \min\limits_{M}; \\ C(M) = \sum\limits_{i} C_{M_i} \le C_{\max}, \end{cases} \tag{7.7}$$

где $P_i \in [0,1]$ – вероятность захвата объектов без применения механизмов защиты;

$M_i = \{0,1\}$ – наличие или отсутствие механизма защиты;

$K_i \in [0,1]$ – коэффициент эффективности механизма защиты M_i;

C_{M_i} – стоимость механизма защиты M_i;

$C_{\max}$ – максимальная совокупная стоимость системы киберзащиты.

В качестве входных данных для решения задачи оптимального размещения механизмов защиты должны подаваться следующие множества:

– множество значений вероятностей захвата объектов системы $P = \{P_1, P_2, ..., P_N\}$, где $P_i = \{P_i^1, P_i^2, ..., P_i^{Ji}\}$ – множество вероятностей захвата *i*-го объекта системы в случае существования нескольких сценариев кибератак (при этом считаем, что кибератаки являются независимыми одна от другой);

– $P_i^j \in [0,1]$ – вероятности захвата;

– множество механизмов защиты со значениями их эффективности и стоимостью $M = \{M_1, M_2, ..., M_N\}$, где $M_i = \{M_{i1}, M_{i2}, ..., M_{iJ_i}\}$ – множество механизмов защиты для защиты от *j*-го сценария атаки *i*-го объекта системы (P_{ij}), при этом считаем, что механизмы защиты являются независимыми один от другого;

– $M_{ij} \in \{0,1\}$ – факт наличия или отсутствия механизма защиты;

– N – количество объектов.

Кроме этого, каждый механизм защиты M_{ij} будет иметь следующие характеристики:

– эффективность механизма защиты – $K \in [0,1]$;

– стоимость механизма защиты – $C_{M_i} \geq 0$.

Исходя из того, что множества механизмов защиты M и коэффициентов эффективности K являются конечными, то данная оптимизационная задача будет относиться к классу задач дискретного программирования, общий вид которых следующий [149]:

$$\max f(x_1, x_2, ..., x_n)$$

при ограничениях

$$g_1(x_1, x_2, ..., x_n) \leq 0;$$

…

$$g_m(x_1, x_2, ..., x_n) \le 0;$$

$$X = [x_1, x_2, ..., x_n] \in D,$$

где D – конечное или счетное множество.

Если одна из функций $f(x_1, x_2, ..., x_n)$ или $g_i(x_1, x_2, ..., x_n) \le 0$ не будет линейной, т. е. не может быть представленной в виде $f(x) = \sum_{i=1}^{n} c_i x_i$, то оптимизационная задача будет относиться к классу задач нелинейной дискретной оптимизации.

Для того чтобы понять, к какому типу принадлежит задача, рассмотрим один из вариантов постановки оптимизационной задачи:

$$\begin{cases} P_{12}(1-M_{12}K_{12})P_3(1-M_3K_3) + P_{11}(1-M_{11}K_{11})(1-P_{12}(1-M_{12}K_{12}))P_3(1-M_3K_3) + \\ +(1-P_{11}(1-M_{11}K_{11}))P_{12}(1-M_{12}K_{12})(1-P_3(1-M_3K_3))P_2(1-M_2K_2) + \\ +P_{11}(1-M_{11}K_{11})P_{12}(1-M_{12}K_{12})P_2(1-M_2K_2)(1-P_3(1-M_3K_3)) + \\ +P_{11}(1-M_{11}K_{11})(1-P_{12}(1-M_{12}K_{12}))P_2(1-M_2K_2)(1-P_3(1-M_3K_3)) \to \min\limits_{M}; \\ M_{11}C_{K_{11}} + M_{12}C_{K_{12}} + M_2C_{K_2} + M_3C_{K_3} \le C_0, \end{cases} \quad (7.8)$$

где $C_{K_{11}}, C_{K_{12}}, C_{K_2}, C_{K_3}$ – стоимость каждого из механизмов защиты;

$C_{\max}$ – максимальная стоимость системы киберзащиты.

Исходя из вида целевой функции $P(P, M, K)$ в приведенном примере (7.8), можно увидеть, что она не будет линейной, и, соответственно, данная задача будет принадлежать к классу задач нелинейной дискретной оптимизации булевого типа.

Методы решения подобных задач делятся на точные (методы отсечения плоскостей, ветвей и границ, последовательного анализа вариантов) и приближенные (метод локальной оптимизации, случайного поиска и эвристические методы). Наиболее распространенными точными методами для решения задач такого типа являются методы, которые реализуют схему последовательного анализа вариантов, метод ветвей и границ и метод отсекающих плоскостей [150].

Таким образом, для получения точного решения задачи (7.7) будем использовать метод последовательного анализа вариантов, приведенный в [151].

7.3. Алгоритм последовательного анализа вариантов для решения задачи оптимального размещения механизмов защиты

Одним из точных методов для решения задачи оптимального размещения механизмов защиты может быть перебор всех возможных вариантов, которые удовлетворяют ограничениям.

Самым простым будет алгоритм, при котором последовательно анализируются все комбинации значений переменных $M = (M_1,...,M_N)$, т. е. вектора:

(1,0,0, ..., 0); (0,1,0, ..., 0); ... ; (1,1, 0, ..., 0); ... ; (1,1,1, ..., 1),

где 1 – если механизм M_i присутствует;

0 – в противоположном случае.

Соответственно, сложность данного алгоритма будет 2^N, где N – количество механизмов защиты. Для уменьшения количества перебираемых вариантов рассмотрим алгоритм последовательного анализа вариантов.

Метод последовательного анализа вариантов основан на конструировании решения шаг за шагом (последовательное уточнение значений компонентов решения) и отбрасывании в процессе конструирования решений, которые не могут быть достроены до оптимального.

Для общего случая задачи дискретной оптимизации схема метода последовательного анализа вариантов будет следующая [152]:

$$\min f(x_1,...,x_n) \tag{7.9}$$

при ограничениях

$$\begin{gathered} g_i(x_1,...,x_n) \le 0, i = \overline{1,m}; \\ x_j \in Q_j,\ j = \overline{1,n}, \end{gathered} \tag{7.10}$$

где $Q_j = \{q_{1j},..,q_{\pi_j j}\}$ – заданные конечные множества.

Вектор $x = (x_1,...,x_n)$ назовем решением, если его компоненты $x_j \in Q_j$, $j = \overline{1,n}$. Множество всех решений обозначим Ω.

Решение является допустимым, если оно удовлетворяет неравенству $g_i(x_1,...,x_n) \le 0$. Множество всех допустимых решений обозначим Ω_f.

Вектор $x_{(p)} = \{x_1,...,x_p\}$, $p < n$, будем называть частичным решением, если $x_j \in Q_j$, $j = \overline{1,p}$. Если он может быть достроен до допустимого решения $(x_1,...,x_p,x_{p+1},...,x_n)$, то будем его называть допустимым частичным решением.

Для того чтобы адаптировать алгоритм решения для конкретной задачи, необходимо указать правило μ выбора частичных решений, которые будут развиваться шаг за шагом (т. е. стратегию последовательного конструирования решений), а также построить набор σ элиминирующих тестов ξ_i для отсева частичных решений, которые не могут быть достроены ни к допустимым, ни к оптимальным. Элиминирующие тесты строятся, исходя из специфики задачи.

Предположим, что в набор σ входят тесты: ξ_0 – анализ допустимости решений и ξ_1 – сравнение допустимых решений по значению целевой функции. Применяя к произвольному множеству решений h (которое содержит как полные, так и частичные решения) тесты ξ_0 и ξ_1, находим в h полные решения и отбрасываем с помощью ξ_0 недопустимые, а с помощью ξ_1 отбрасываем те из допустимых решений, для которых в h существуют лучшие решения по значению целевой функции. Но если применяем только тесты $\sigma = \{\xi_0, \xi_1\}$, то получаем алгоритм полного перебора, приведенный вначале.

Применим этот подход для решения оптимизационной задачи

$$\begin{cases} P(M) = P(M_1,...,M_N) = P(P,M,K) \to \min\limits_{M}; \\ C(M) = \sum\limits_i C_{M_i} \le C_{\max}, \end{cases} \tag{7.11}$$

Целевая функция $P(P,M,K)$, и ограничение $g = \sum\limits_{i=1..N} C(M) - C_{\max} \le 0$ являются монотонными по всем переменным. Монотонность целевой функции следует из утверждения 2, которое было доказано в п. 5.7.

Для упрощения алгоритма задачу оптимального синтеза системы кибернетической защиты информации будем рассматривать для случая, когда для противодействия захвату объекта может использоваться лишь один механизм защиты.

Тогда алгоритм решения задачи состоит из следующих этапов:

A. Нахождение начального значения рекорда.

B. Сортировка переменных для более эффективного поиска.

C. Построение дерева поиска.

Рассмотрим каждый из этапов детальнее.

Этап А.

Начальное значение рекорда P^* – оценка сверху для минимума целевой функции. Для его определения находим некоторое допустимое решение $M^* = \left(M_1^*,...,M_N^*\right)$, $M_i^* \in \{0,1\}$, и значение целевой функции (функции вероятности успешности кибератаки) на нем принимаем за значение рекорда: $P^* = P(M_1^*,...,M_N^*)$.

M^* строим следующим образом:

Шаг 0. $M^* = (0,...,0)$;

$i = 1$.

Шаг 1. Если $i = N+1$, то завершаем поиск, причем полученное решение является оптимальным.

Иначе $M_i^* = 1$.

Шаг 2. Если $g(M_1,...,M_N) \le 0$ (т. е. полученное решение является допустимым), то $i = i+1$, и возвращаемся на Шаг 1.

Иначе, $M_i^* = 0$. Завершаем поиск.

Этап В.

Сортировку выполняем или в порядке уменьшения стоимости механизмов защиты, или в порядке роста «полезности» механизмов защиты. В первом случае сортировка направлена на то, чтобы при построении дерева поиска сразу же отбросить слишком дорогие механизмы защиты. Во втором случае стараемся отбросить те механизмы защиты, которые незначительно увеличивают защищенность сети. «Полезность» механизмов защиты $U(K_i)$ вычисляется по формуле

$$U(K_i) = \sum_{j=1..L} \frac{\partial P(E_1,...,E_N,0,...,0)}{\partial E_j} \cdot K_{ij}, \qquad (7.12)$$

где K_{ij} – эффективность *i*-го механизма защиты против j-й кибератаки;

L – количество кибератак.

Таким образом, в $U(K_i)$ учитывается значимость кибератак, против которых направлен механизм защиты, и эффективность самого механизма защиты.

Этап С.

В соответствии с методом последовательного анализа вариантов, строим дерево поиска на частичных решениях задачи.

Выполняем обход дерева в ширину и на каждом этапе развиваем только перспективные частичные решения. Для определения перспективности решения вычисляем его стоимость $g(M_1^0,...,M_{depth}^0,0,...,0)$ и подсчитываем значения целевой функции $P(M_1^0,...,M_{depth}^0,1,...,1)$. Кроме того, учитываем, что для частичных решений вида $\{M_1^0,...,M_n^0,0\}$ оценка стоимости будет такая же, как для родительского решения $\{M_1^0,...,M_n^0\}$. Аналогично, для частичных решений вида $\{M_1^0,...,M_n^0,1\}$ оценка значения целевой функции будет такая же, как $\{M_1^0,...,M_n^0\}$, что позволяет не вычислять их повторно.

Шаг 0. $depth = 1$. Помещаем в очередь частичные решения $M_{(1)}^0 = \{0\}$ и $M^0{}_{(1)} = \{1\}$.

Шаг 1. Вынимаем из очереди первый элемент – частичное решение $M_{(depth)}^0$.

Если $depth == N$ (построили полное решение), то переходим на Шаг 6.

Шаг 2. Если $M_{depth}^0 = 0$ (последний элемент частичного решения равняется нулю), то переходим на Шаг 3.

Иначе переходим на Шаг 4.

Шаг 3. Оценка значения целевой функции на частичном решении.

Если $P(M_1^0,...,M_{depth}^0,1,...,1) \le P^*$ (частичное решение можно достроить до оптимального), то помещаем в конец очереди частичные решения $M_{(depth+1)}^0 =$ $= \{M_1^0,...,M_{depth}^0,0\}$, $M_{(depth+1)}^0 = \{M_1^0,...,M_{depth}^0,1\}$.

Переходим на Шаг 5.

Шаг 4. Оценка значения функции ограничений на частичном решении.

Если $g(M_1^0,...,M_{depth}^0,0,...0) \le 0$ (частичное решение можно достроить до допустимого), то помещаем в конец очереди частичные решения $M_{(depth+1)}^0 =$ $= \{M_1^0,...,M_{depth}^0,0\}$, $M_{(depth+1)}^0 = \{M_1^0,...,M_{depth}^0,1\}$.

Шаг 5.

Если $g(M_1^0,...,M_{depth}^0,0,...,0) \le 0$, то $P^* = \min\{P(M_1^0,...,M_{depth}^0,0,...,0), P^*\}$.

Переходим на Шаг 1.

Шаг 6. Оценка значения функции ограничений и целевой функции на полном решении.

Если $g(M_1^0,...,M_N^0) \le 0$ (решение является допустимым) и $P(M_1^0,...,M_N^0) \le P^*$, то $P^* = P(M_1^0,...,M_N^0)$ и $M^* = M_N^0$.

Переходим на Шаг 1.

Для проверки работоспособности предложенного алгоритма последовательного анализа вариантов применим его для задачи (7.8).

Определим оптимальные механизмы защиты, предположив, что их эффективность $K_i = 1$, а максимальная совокупная стоимость системы кибер-защиты $C_{\max} = 5$. Полученные результаты приведены в табл. 7.1.

Таблица 7.1

Результаты тестирования алгоритма последовательного анализа вариантов (ПАВ)

	Тест 1		**Тест 2**	
	Вероятность захвата	Стоимость механизмов	Вероятность захвата	Стоимость механизмов
P_2	0,5	3	0,5	3
P_3	0,4	3	0,6	3
P_{11}	0,8	2	0,8	2
P_{12}	0,9	4	0,9	4
	P = 0,36; Механизмы М11 и М2		P = 0,45; Механизмы М11 и М3	

Полученные результаты тестов показывают работоспособность разработанного алгоритма последовательного анализа вариантов.

7.4. Процедура оптимального размещения механизмов защиты ИКС

Разработаем процедуру для оптимального размещения механизмов защиты в узлах ИКС.

1. Входными данными является граф кибератаки $G(V, E, A, O, P)$, где для каждого объекта системы v_i задана вероятность его захвата P_i (табл. 7.2).

2. Записывается функция вероятности успешности кибератаки (7.5), с учетом механизмов защиты и коэффициентов эффективности $P(P, M, K)$.

3. При фиксированной топологии сети G^* к графу кибератак добавляются механизмы защиты и строится граф киберзащиты $G(V, E, A, O, P, M, K)$, где для каждого механизма защиты также указывается его стоимость C_{M_i} (табл. 7.2).

4. С помощью алгоритма последовательного анализа вариантов решается оптимизационная задача (7.11). В случае нескольких вариантов механизмов защиты для одного объекта алгоритм последовательного анализа вариантов должен быть запущен для всех комбинаций.

5. Результатом работы процедуры является множество механизмов защиты M^* и значение функции вероятности успешности кибератаки $P_{res} = P(G^*, M^*)$.

Таблица 7.2

Механизмы защиты для защиты объектов ИКС

Объекты системы	Вероятности захвата	Механизмы защиты
v_1	P_1	(M_1, K_1, C_{M_1})
v_2	P_2	$(M_{2-1}, K_{2-1}, C_{M_{2-1}})$; $(M_{2-2}, K_{2-2}, C_{M_{2-2}})$
v_3	P_3^1	$(M_{31}, K_{31}, C_{M_{31}})$
	P_3^2	$(M_{32-1}, K_{32-1}, C_{M_{32-1}})$; $(M_{32-2}, K_{32-2}, C_{M_{32-2}})$
	…	…
…	…	…
v_i	…	…
	P_i^j	$(M_{ij-1}, K_{ij-1}, C_{M_{ij-1}})$; $(M_{ij-2}, K_{ij-2}, C_{M_{ij-2}})$; …; $(M_{ij-k}, K_{ij-k}, C_{M_{ij-k}})$
	…	…
…	…	…
v_N	…	…

7.5. Пример оптимального размещения механизмов защиты в ИКС и анализ полученных результатов

Рассмотрим синтез оптимальной системы защиты для примера ИКС, приведенной на рис. 7.3. Данная система содержит семь объектов $V = \{v_1, ..., v_7\}$.

На рис. 7.3 приведены источники угроз v_1, v_4, v_5 и объект кибератаки v_7 – сервер Баз данных. На основе этих данных и топологии сети запишем функцию вероятности успешности кибератаки:

$$P = P_4 + P_1P_2(1-P_4) + P_1P_3(1-P_2)(1-P_4) + P_6P_5(1-P_1)(1-P_4) + P_1P_6P_5(1-P_2)(1-P_3)(1-P_4).$$

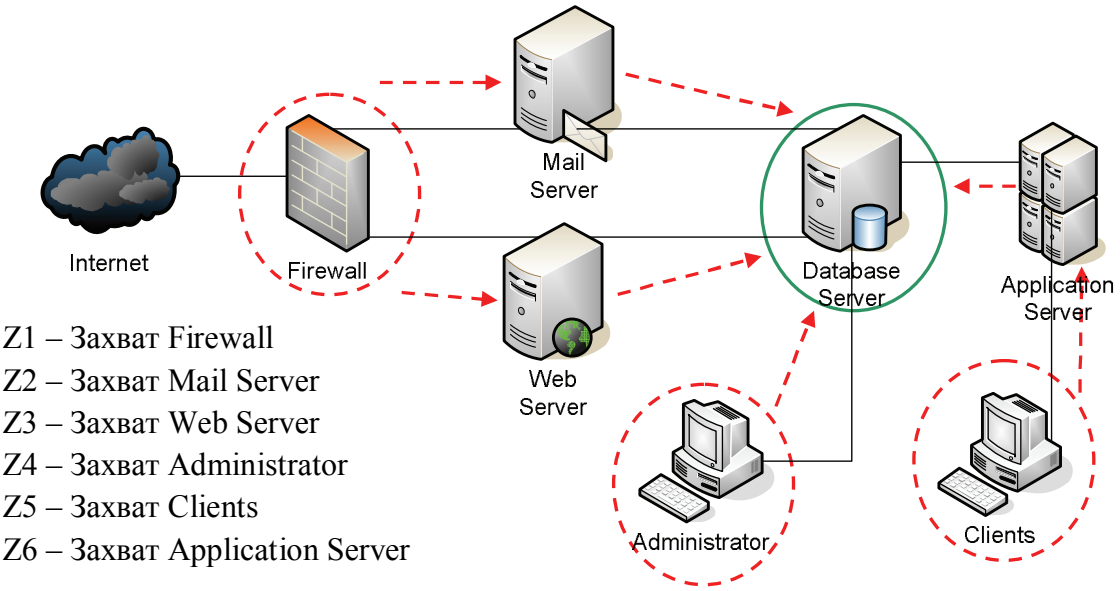


Рис. 7.3. Пути захвата сервера Баз данных

Предполагая, что для каждого из объектов существует один вариант его захвата, запишем данную функцию с использованием вероятностей захвата P, механизмов защиты M и их коэффициентов эффективности K:

$$P(P,M,K) = P_4(1-M_4K_4) +$$
$$+ P_1(1-M_1K_1)P_2(1-M_2K_2)(1-P_4(1-M_4K_4)) +$$
$$+ P_1(1-M_1K_1)P_3(1-M_3K_3)(1-P_2(1-M_2K_2))(1-P_4(1-M_4K_4)) +$$
$$+ P_6(1-M_6K_6)P_5(1-M_5K_5)(1-P_1(1-M_1K_1))(1-P_4(1-M_4K_4)) +$$
$$+ P_1(1-M_1K_1)P_6(1-M_6K_6)P_5(1-M_5K_5)(1-P_2(1-M_2K_2))(1-P_3(1-M_3K_3))(1-P_4(1-M_4K_4)).$$

Задача синтеза оптимальной системы киберзащиты будет состоять в выборе механизмов защиты таким образом, чтобы они минимизировали значение функции вероятности при ограничении на их совокупную стоимость.

Зададим табл. 7.3 со значениями вероятностей захвата объектов, а также табл. 7.4 с механизмами защиты, коэффициентами эффективности и стоимостью. Максимальная совокупная стоимость системы киберзащиты не должна превышать $C_{\max} = 5$.

Эти данные являются входными для алгоритма последовательного анализа вариантов. Количество шагов работы данного алгоритма и процесс его сходимости приведены в табл. 7.5.

В первой колонке табл. 7.5 приведены номера механизмов защиты, которые могут быть использованы, во второй – вероятности захвата объектов, в третьей – стоимость механизмов защиты и в четвертой – коэффициенты

эффективности механизмов. Дальше по шагам показана работа алгоритма последовательного анализа вариантов: 1 – если рекомендуется применять соответствующий механизм; 0 – не рекомендуется. Алгоритм сходится через 28 шагов, что более чем в 2 раза быстрее, чем при использовании алгоритма полного перебора всех вариантов (64 варианта).

Таблица 7.3

Угрозы для объектов ИКС и вероятности их реализации

	P	**Описание**
P_1	0,6	нахождение ошибок в настройках и подбор пароля администратора межсетевого экрана
P_2	0,2	заражение вирусом
P_3	0,4	SQL инъекция или Cross Site Scripting (XSS)
P_4	0,7	подбор пароля администратора
P_5	0,1	заражение вирусом
P_6	0,5	слабая авторизация

Таблица 7.4

Механизмы защиты, их коэффициенты эффективности и стоимость

M	**К**	**С**	**Описание**
M_1	0,5	3	Установить стойкий пароль и соединение SSL
M_2	0,3	1	Web Application Firewall
M_3	0,1	2	Тестирование
M_4	0,4	2	Карточная система доступа к компьютерам
M_5	0,3	4	Использование антивируса
M_6	0,6	1	Авторизация по электронным сертификатам

Таблица 7.5

Процесс сходимости алгоритма последовательного анализа вариантов

M	P	C	K	1	2	3	4	5	6	7	8	9	10	11	12	13	14
5	0,1	4	0,3	1	0	1	1	0	0	1	1	0	0	0	0	1	1
1	**0,6**	**3**	**0,5**			**1**	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
3	0,4	2	0,1							1	0	1	0	1	0	0	0
4	**0,7**	**2**	**0,4**													**1**	**0**
2	0,2	1	0,3														
6	0,5	1	0,6														

M	P	C	K	15	16	17	18	19	20	21	22	23	24	25	26	27	28
5	0,1	4	0,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	**0,6**	**3**	**0,5**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
3	0,4	2	0,1	1	1	0	0	1	1	0	0	1	1	0	0	0	0
4	**0,7**	**2**	**0,4**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
2	0,2	1	0,3									1	0	1	0	0	0
6	0,5	1	0,6													1	0

Таким образом, при ограничении на стоимость системы киберзащиты $C_{max} = 5$ получаем, что оптимальным является использование механизмов защиты на межсетевом экране (M_1) и на сервере администратора (M_4), при этом значение функции вероятности успешности кибератаки уменьшается с $P = 0{,}80392$ до $P = 0{,}555256$.

Такое расположение механизмов защиты также отвечает практическому опыту построения систем защиты.

Выводы

Для решения задачи оптимального размещения механизмов защиты в узлах ИКС в функцию вероятности успешности кибератак введены механизмы защиты и их характеристики – коэффициент эффективности.

На основе метода последовательного анализа вариантов разработана процедура оптимального размещения механизмов защиты при ограничении на их суммарную стоимость (стоимость системы киберзащиты) и при фиксированной топологии сети.

8. СИНТЕЗ КИБЕРНЕТИЧЕСКОЙ СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ С ОПТИМАЛЬНЫМИ ХАРАКТЕРИСТИКАМИ

Наряду с задачами определения экстремальных показателей уровня киберзащищенности также рассматривают задачи определения экстремальных экономических показателей систем кибернетической защиты. Эти показатели определяются стоимостью механизмов защиты, которые были выбраны на этапе проектирования соответствующей КС КЗИ, затратами на преодоление последствий успешных кибератак злоумышленников, накладными и другими затратами для обеспечения функционирования КС КЗИ. Конечной целью синтеза систем такого класса является нахождение экстремума экономических показателей в соответствующей ИКС при учете ограничений на значение уровня киберзащищенности.

Среди задач определения экстремальных экономических показателей важное практическое значение имеет задача нахождения такой конфигурации механизмов защиты с минимальной стоимостью, которая обеспечивает гарантированный остаточный риск в ИКС (или гарантированный уровень киберзащищенности). Разработка КС КЗИ в такой постановке и является целью данной главы.

8.1. Концепции обеспечения безопасности структурно сложных систем

Существуют несколько концепций построения систем обеспечения минимума риска (безопасности): «нулевого риска», «ненулевого риска» и смешанный подход [153]. Концепция «нулевого риска» имеет другое название – детерминистский подход при обеспечении минимума риска. В ее основе лежит предположение, что можно исключить любую опасность, если не пожалеть денег на построение системы минимизации риска и обеспечить высокий уровень дисциплины. Эта концепция является неадекватной по внутренним законам бизнеса, функционированию банков и сложных технических систем. Упомянутые законы имеют вероятностный характер, и риск всегда существует. Концепция «ненулевого риска», или концепция «приемлемого риска», базируется на численной оценке вероятности финансовой или технической катастрофы. Вероятностный анализ и управление рисками позволяет применить

множество новых мер для повышения безопасности и снижения риска, целенаправленно концентрировать и перераспределять средства не только для предупреждения финансовых или технических катастроф, но и для заблаговременной подготовки к действиям в экстремальных условиях. Под численным измерением риска понимают или определение вероятности сложного события, инициируемого элементарными событиями, или вычисление средних потерь (усреднение происходит с помощью вычисления суммы всех потенциальных потерь с весами, пропорционально вероятностям возникновения этих потерь). После численного измерения риска возможно установить его допустимые границы. Отсутствие методов численного измерения риска приводит к тому, что решения, принимаемые для повышения безопасности и снижения уровня риска, являются неопределенными по своему характеру [98].

Существует большой опыт разработки моделей риска, безопасности и применения логико-вероятностного метода в атомной индустрии, морском флоте и др. Такие системы являются структурно сложными и включают в себя оборудование, компьютерные и программные средства, действия персонала по обслуживанию, тестированию и управлению. Для многих современных систем риск стал нормальным явлением, так как их функционирование по своей природе связано с возникновением риска (таковы, например, банковские системы, в которых всегда есть риск невозврата кредитов). Для того чтобы моделировать и оценивать безопасность реального мира, необходимы различные модели и мощные информационные технологии. Наряду с применением логико-вероятностного метода для исследования технических систем также существуют и эффективно используются модели банковских рисков и мошенничества в бизнесе.

Информация, обрабатываемая в современных киберсистемах, всегда имеет определенную ценность, хотя очень часто ее трудно определить количественно. Нулевой риск можно было бы обеспечить, если бы все киберугрозы информации, возникающие в процессе ее функционирования, эффективно нейтрализовались средствами защиты, которые были включены в соответствующую КС КЗИ. На практике достижение нулевого уровня риска не является реальным. Поскольку из соображений здравого смысла стоимость КС КЗИ не должна превышать максимальные потери от реализации киберугроз, то обычно считается, что стоимость КС КЗИ должна составлять не более определенного процента от потерь в худшем случае.

Управление рисками в ИКС имеет свои особенности, которые описаны в международных стандартах в области защиты информации [72, 73, 154]. В упомянутых стандартах было введено понятие об остаточном риске, который является интегральным показателем, отражающим величину потенциальных потерь, которые могут возникнуть после успешной реализации кибератаки, и может рассматриваться как один из критериев качества киберсистем. При проектировании такого класса структурно сложных технических систем, как современные КС КЗИ, работа должна вестись именно в направлении минимизации или ограничения величины риска. В киберсистемах, принадлежащих государственным учреждениям, на построение КС КЗИ может выделяться определенный бюджет, соответственно накладывается ограничение на совокупную стоимость использованных средств защиты, а вероятность нарушения безопасности минимизируется. В системах, эксплуатируемых в коммерческих структурах, целью может быть построение КС КЗИ с минимальной стоимостью при наложении ограничения на величину риска.

8.2. Функция потерь реализации кибератаки

Для наиболее распространенного стека протоколов TCP/IP количество уровней модели взаимодействия открытых систем сокращается до четырех – уровень доступа к сети (или нижний), сетевой (или уровень «Интернет»), транспортный и уровень приложений. На рис. 8.1 приведен схематический вид КС КЗИ, в которой некоторое количество механизмов защиты на каждом уровне стека противодействует реализации угроз, уменьшая тем самым уровень остаточного риска. В предыдущих пунктах было показано, что количество уровней стека не влияет на общую структуру математической модели, поэтому в дальнейшем будет рассматриваться абстрактный стек с количеством уровней N (рис. 8.1) [32].

Имеем следующее выражение для вероятности нарушения безопасности на уровнях стека $\{i_1, i_2, \ldots, i_s\} \subset W$ в результате воздействия одной киберугрозы:

$$P_{i_1,\ldots,i_s} = \prod_{i_k \in \{i_1,\ldots,i_s\}} p_{i_k} \cdot \prod_{i_k \in W \setminus \{i_1,\ldots,i_s\}} (1 - p_{i_k}) \cdot \prod_{k=1}^{\min(i_1,\ldots,i_s)} (1 - q_k),$$

где $W = \{1,2,\ldots,N\}$;

N – количество уровней стека протоколов;

p_i – вероятность нанесения ущерба на i-м уровне из-за реализации кибер-угрозы;

q_i – вероятность исключить негативное воздействие угрозы на i-м уровне.

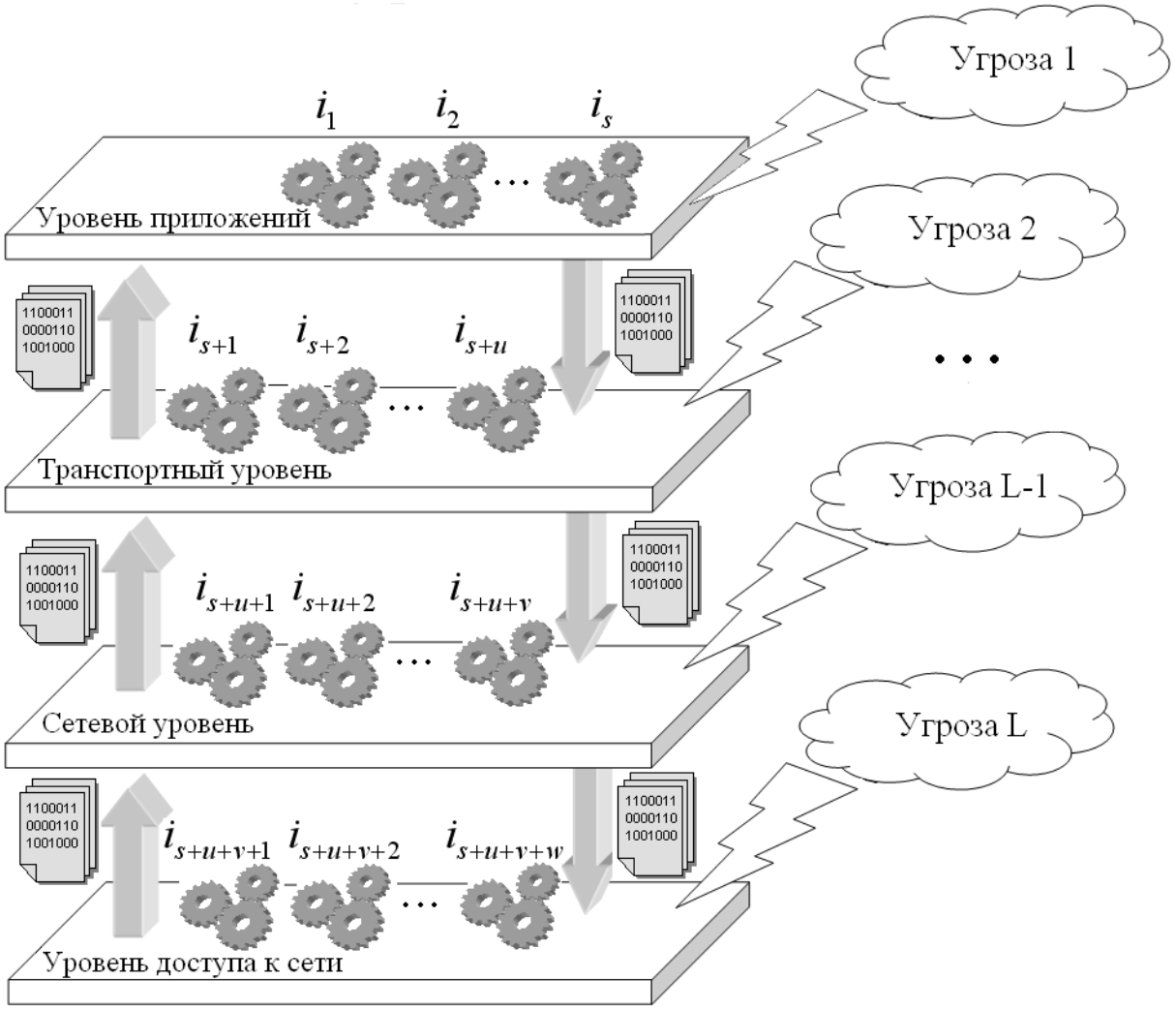


Рис. 8.1. Архитектура КС КЗИ в открытой ИКС

Исходя из основных принципов логико-вероятностного подхода для описания условий функционирования структурно сложных систем, было получено общее выражение для функции опасности для абстрактного стека протоколов. Далее с использованием той же методологии будет получено выражение для средних потерь от реализации кибератаки. Пусть существуют некая киберугроза и механизм, который эффективно ее нейтрализует. Будем считать, что величина q_i принимает значение 1, если на i-м уровне находится механизм защиты, и 0 – в противном случае. Таким образом, механизм защиты будет считаться абсолютно надежным. Для учета потерь от реализации

киберугрозы на i-м уровне введем коэффициент Q_i. Таким образом, для стека с N уровнями выражение для средних потерь может быть получено в виде

$$Q = \sum_{s=1}^{N} \sum_{\substack{i_1, i_2, \ldots, i_s \subset W \\ i_1 < i_2 < \ldots < i_s}} P_{i_1, i_2, \ldots, i_s} \cdot \sum_{k=1}^{s} Q_{i_k} .$$

Функция Q после упрощений и подстановок будет иметь вид

$$Q = \sum_{j=1}^{N} R_j \prod_{k=1}^{j-1} (1 - R_k) \prod_{k=1}^{j} (1 - M_k) \left(Q_j + \sum_{k=j+1}^{N} R_k Q_k \right), \qquad (8.1)$$

где сделано переобозначение $R_i = p_i$ и $M_i = q_i$ для простоты сравнения с функцией опасности. Выражение (8.1) получено индуктивным путем.

Полученное соотношение является частичным случаем, справедливым тогда, когда каждый механизм устраняет единственную киберугрозу. В общем случае каждый механизм может быть эффективно использован для устранения нескольких киберугроз. Следует также заметить, что для этого случая допускается полный отказ от реализации механизма защиты. В противном случае решение задачи упрощается, и он в работе не рассматривается.

Поскольку риск является аддитивной величиной для независимых угроз, то переход к общему случаю, когда рассматриваются L киберугроз, проводим путем добавления в соотношении (8.1) к переменным индекса i-го номера киберугрозы и суммирования по этому индексу:

$$Q_L = \sum_{i=1}^{L} \left[\sum_{j=1}^{N} \left(R_{ij} \prod_{k=1}^{j-1} (1 - R_{ik}) \prod_{k=1}^{j} (1 - M_{ik}) \left(Q_{ij} + \sum_{k=j+1}^{N} R_{ik} Q_{ik} \right) \right) \right], \qquad (8.2)$$

где L – количество киберугроз;

N – количество уровней стека протоколов;

R_{ij} – вероятность реализации i-й угрозы на j-м уровне;

Q_{ij} – величина потерь от реализации i-й киберугрозы на j-м уровне;

M_{ij} – переменная, отображающая факт реализации i-го механизма защиты на j-м уровне;

Q_L – средние потери от реализации L киберугроз.

Выражение (8.2) для средних потерь от нарушения безопасности может быть использовано для синтеза КС КЗИ, если нам известны коэффициенты R_{ij}, Q_{ij}. На практике их определение является отдельной проблемой, поэтому

будем использовать подход, который опирается на методы математической статистики и принятия решений (метод анализа иерархий в частности).

8.3. Синтез кибернетической системы защиты информации

С учетом возможности описания функционирования КС КЗИ с открытой архитектурой с помощью логико-вероятностных методов и полученной зависимости (8.2) для средних потерь от реализации L киберугроз в частности, задача минимизации стоимости КС КЗИ при ограничении на величину остаточного риска будет выглядеть следующим образом:

$$\begin{cases} \sum_{i=1}^{L}\left[\sum_{j=1}^{N}\left(R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\prod_{k=1}^{j}\overline{M}_{ik}\left(Q_{ij}+\sum_{k=j+1}^{N}R_{ik}Q_{ik}\right)\right)\right] \le R_{\max} \\ \sum_{i=1}^{L}\sum_{j=1}^{N}\overline{M}_{ij}C_{ij} \to \max\limits_{\overline{M}_{ij}\in\{0,1\}} \end{cases}, \qquad (8.3)$$

где $R_{\max}$ – верхняя граница средних потерь от реализации кибератаки;

C_{ij} – стоимость реализации механизма защиты от i-й киберугрозы на j-м уровне;

$\overline{M}_{ij}$ – дополнение M_{ij} до единицы.

Полученная задача принадлежит к классу задач булевого программирования с линейной целевой функцией и нелинейным ограничением. Задачи булевого программирования, в которых встречаются нелинейности вида произведения, с помощью введения вспомогательных переменных и добавления к множеству начальных ограничений по одной паре дополнительных для каждой новой переменной сводятся к линейным. Для задачи синтеза КС КЗИ с минимальной стоимостью механизмов защиты, которая при этом будет гарантировать уровень средних потерь не выше заданной границы, такой подход является допустимым, но особенности задачи позволяют сделать еще большие упрощения.

Если для нейтрализации киберугрозы i выбирается реализация механизма защиты на j-м уровне стека протоколов, а на всех предыдущих уровнях такая реализация не производится, то на старших уровнях реализация уже является лишней, так как риск от этого не изменяется (все механизмы

считаются абсолютно надежными), а стоимость увеличивается. Таким образом, среди всех 2^N возможных двоичных векторов $\{\overline{M}_{i1},\ldots,\overline{M}_{iN}\}$, $i=i_0$, в оптимальное решение могут входить только $N+1$, а именно:

$$\{\underbrace{1,\ldots,1}_{k},0,\underbrace{1,\ldots,1}_{N-k-1}\},\ 0\le k<N \quad \text{и} \quad \{1,\ldots,1\}.$$

При условии $\sum_{i=1}^{N}\overline{\xi}_i \ge N-1,\ \overline{\xi}_i\in\{0,1\}$ (все приведенные выше векторы удовлетворяют этому условию) справедливым является равенство $\prod_{i=1}^{s}\overline{\xi}_i = 1-\sum_{i=1}^{s}\xi_i$, так как все перекрестные произведения исчезают. Поэтому выражение для ограничения задачи (8.3) можно переписать в виде:

$$\sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=1}^{j}M_{ik}R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\left(Q_{ij}+\sum_{k=j+1}^{N}R_{ik}Q_{ik}\right)\ge R^*,$$

$$R^*=\sum_{i=1}^{L}\sum_{j=1}^{N}\left(R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\left(Q_{ij}+\sum_{k=j+1}^{N}R_{ik}Q_{ik}\right)\right)-R_{\max}.$$

Замечаем также, что

$$\sum_{j=1}^{N}\sum_{k=1}^{j}M_{ik}R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\left(Q_{ij}+\sum_{k=j+1}^{N}R_{ik}Q_{ik}\right)=$$

$$=\sum_{j=1}^{N}M_{ij}\left(\sum_{k=j}^{N}R_{ik}Q_{ik}\prod_{k=1}^{j-1}(1-R_{ik})\right),$$

$$\sum_{i=1}^{L}\sum_{j=1}^{N}R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\left(Q_{ij}+\sum_{k=j+1}^{N}R_{ik}Q_{ik}\right)=\sum_{i=1}^{L}\sum_{j=1}^{N}R_{ij}Q_{ij},$$

поэтому окончательный вид эквивалентной задачи линейного булевого программирования будет следующим:

$$\begin{cases}\sum_{i=1}^{L}\sum_{j=1}^{N}M_{ij}\sum_{k=j}^{N}R_{ik}Q_{ik}\prod_{k=1}^{j-1}(1-R_{ik})\ge\sum_{i=1}^{L}\sum_{j=1}^{N}R_{ij}Q_{ij}-R_{\max},\\ \sum_{j=1}^{N}M_{ij}\le 1,\ \ i=\overline{1,L}, \qquad \sum_{i=1}^{L}\sum_{j=1}^{N}M_{ij}C_{ij}\to\min\limits_{M_{ij}\in\{0,1\}}.\end{cases} \tag{8.4}$$

Для решения задачи (8.4) можно применять точные методы линейного булевого программирования – метод Балаша [155] (в т. ч. модифицированный [156]), метод ПАВ [149; 157], P-метод [158] и др.). Но, учитывая то, что размерность задачи (8.4) и количество ограничений в ней можно уменьшить,

перейдя к эквивалентной задаче сепарабельного программирования, был сделан переход к задаче:

$$\begin{cases} J = \sum_{i=1}^{L} f_i(x_i) \to \min\limits_{x_i \in X_i}, \\ \sum_{i=1}^{L} g_i(x_i) \le R_{\max}, \end{cases} \tag{8.5}$$

где $x_i = s,\ s = \overline{1, N+1}$ отвечает реализации i-го механизма защиты на s-м уровне (фиктивный уровень $N+1$ означает отказ);

f_i – стоимость реализации i-го механизма защиты на соответствующем уровне;

g_i – остаточный риск, который остается после реализации механизма защиты и равен слагаемому для фиксированного i из ограничения задачи (8.3).

Вместо ограничения задачи (8.4) рассматривалось эквивалентное ограничение из задачи (8.3). С переходом к задаче сепарабельного программирования были потеряны некоторые особенности структуры основного ограничения задачи (8.4), а вместо этого мы получили таблично заданную зависимость.

Заметим, что для решения практических задач синтеза КС КЗИ использование точных методов не является обязательным, а иногда и неоправданным, в частности последнее имеет место в том случае, когда численные характеристики механизмов защиты и потенциальных киберугроз известны неточно. Единственным точным параметром в предложенной математической модели является стоимость механизмов защиты, а остальные определяются экспертным путем или статистически. В данном пункте используется метод вектора спада [150], который принадлежит к классу методов локальной дискретной оптимизации. Методы локальной оптимизации вообще, и метод вектора спада в частности, не требуют выполнения жестких условий на вид целевой функции или ограничений и отличаются высокой скоростью работы. Заметим, что полученную задачу (8.5) также можно решать с помощью метода ПАВ.

При использовании методов локальной оптимизации важным является выбор начальной точки, с которой начинается поиск оптимального решения. Эта точка может выбираться как случайно (конечно, при этом она должна удовлетворять ограничению), так и согласно неким эвристическим правилам. Второй вариант поясним следующим образом. Поскольку использование

механизма защиты на первом (самом нижнем) уровне стека протоколов снижает потери от определенной киберугрозы до нуля, а отказ от использования механизма не увеличивает стоимость системы, то именно из этих крайних вариантов можно выбирать некоторые конфигурации механизмов защиты (назовем такие конфигурации «пессимистическими»). Наилучшая конфигурация из всего множества «пессимистических» не обязательно будет оптимальной в описанном ранее смысле, но все же может быть использована в качестве начальной точки для алгоритма вектора спада. Таким образом, для решения задачи сепарабельного программирования с помощью метода вектора спада начальную точку можно найти с помощью решения вспомогательной задачи линейного программирования:

$$\begin{cases} \sum_{i=1}^{L} z_i C_{i1} \to \max\limits_{0 \le z_j \le 1}, \\ \sum_{i=1}^{L} z_i W_{i,N+1} \le R_{\max}, \end{cases} \tag{8.6}$$

где $W_{i,N+1} = \sum_{j=1}^{N} R_{ij} Q_{ij}$ является той составной частью общего риска, которая возникает при отказе от реализации механизма защиты от i-й киберугрозы;

C_{i1} – стоимость реализации механизма на первом уровне стека протоколов.

Известно [159], что задача (8.6) имеет тривиальное решение после переименования переменных. В случае, когда выполняется

$$\frac{C_{11}}{W_{1,N+1}} \ge \frac{C_{21}}{W_{2,N+1}} \ge \dots \ge \frac{C_{L1}}{W_{L,N+1}},$$

решение имеет вид $\{z_1^*, z_2^*, \dots, z_L^*\}$, где

$$z_i^* = \begin{cases} 1, \text{ если } i \le r_0, \\ 0, \text{ если } i > r_0 + 1, \end{cases} \quad z_{r_0+1}^* = \frac{R_{\max} - \sum_{i \le r_0} W_{i,N+1}}{W_{r_0+1,N+1}},$$

где $r_0 = \max\{r \mid \sum_{i \le r} W_{i,N+1} \le R_{\max}\}$.

После уменьшения значения координаты $z_{r_0+1}^*$ до 0 мы получим целочисленное решение, которому отвечает некоторое решение начальной задачи (оно является допустимым и близким к оптимуму среди «пессимистических» конфигураций). Связь между ними является следующей: если $z_i^* = 1$, то $x_i^* = N + 1$, иначе $x_i^* = 1$.

Для возможности в дальнейшем ссылаться на термины, используемые при описании алгоритма вектора спада, введем следующие обозначения:

$\rho(x, y) = \sum_{i=1}^{L} |x_i - y_i|$ – метрика в пространстве решений задачи (8.5);

$B_n(P) = \{X : \rho(P, X) \le n\}$ – окрестность радиуса n решения P.

Таким образом, окончательно алгоритм решения задачи синтеза КС КЗИ имеет вид:

1. Если возможно сделать перебор всех 2^L конфигураций КС КЗИ, которые включают в себя только реализацию механизма на самом нижнем уровне стека ($x_i = 1$), или отказ от использования ($x_i = N + 1$), то сделать этот перебор и сохранить из них не более K наилучших по значению критерия J. Если нет, то генерировать на протяжении интервала времени T случайные конфигурации, которые удовлетворяют неравенству из (8.5), и сохранить после его завершения не более K вариантов с наилучшим значением критерия J в списке O.

2. Если полный перебор не делался, то найти конфигурацию КС КЗИ, которая отвечает решению вспомогательной задачи линейного программирования (8.6), и добавить его к уже известным стартовым точкам в O.

3. Для всех точек $P \in O$ найти локальный минимум относительно $B_3(P)$ с помощью метода вектора спада, отсортировать все стартовые точки по $J(\mu_P)$, где μ_P – локальный минимум, в который мы попадаем из соответствующей точки P. Сохранить отсортированный список S с парами «стартовая точка/локальный минимум» для возможности отката. Удалить из S пару $(P^*, \mu_{P^*}) = \arg\min J(\mu_P)$.

4. Оцениваем устойчивость локального минимума μ_{P^*} путем повторного поиска по алгоритму вектора спада после небольших возмущений стартовой точки. Если существует локальный минимум $\mu_{P'} \neq \mu_{P^*}$ для $P' \in B_{r'}(P^*)$ и некоторого r', то удалить из списка S следующую по значению критерия J пару (P^*, μ_{P^*}) и вернуться к шагу 4, иначе возвратить P^* – оптимальную конфигурацию СЗИ.

8.4. Пример синтеза кибернетической системы защиты информации с минимальной стоимостью механизмов защиты

Для исследования эффективности решения задачи синтеза КС КЗИ в приведенной постановке была решена модельная задача с количеством механизмов защиты (соответственно, и количеством киберугроз) $L = 20$ для стека протоколов TCP/IP с количеством уровней $N = 4$. Матрицы Q_{ij}, R_{ij} и C_{ij} были сгенерированы случайным образом (только дополнительно налагалось условие $C_{ik} > C_{il}$ для $k > l$, так как противоположный случай нетипичен).

Начальные точки, из которых начинался поиск, находились двумя способами. Одним из таких способов был поиск K наилучших решений среди векторов X с координатами $x_i \in \{0, N\}$ (т. е. область значений каждой координаты сужалась к крайним значениям) и использование их в качестве начальных точек. Для примера, который рассматривался, полный перебор 2^L всех «ограниченных» решений был еще приемлемым, поэтому генерирование случайных решений не использовалось.

Дополнительная точка, из которой также начинался поиск, находилась с помощью решения упомянутой выше вспомогательной задачи линейного программирования (8.6). Поскольку делался полный перебор, то это не давало нам новую стартовую точку, но в общем случае, когда это невозможно, это решение задачи может быть эффективно использовано.

Для реализации метода вектора спада была выбрана приведенная выше метрика, а поиск совершался в окрестности радиуса 3 (увеличение радиуса до четырех оказалось неоправданным, так как не приводило к улучшению уже полученного решения). Согласно [150], переход к лучшей точке происходил сразу после ее нахождения, т. е. абсолютный минимум в окрестности не находился, а переходили к первой допустимой точке, которая улучшала значение целевой функции. В окрестности радиуса 1 поиск лучшей точки происходил случайным образом, т. е. все возможные направления изменения каждый раз перебирались в другом порядке, а для окрестностей радиусов 2 и 3 ограничились последовательным перечислением направлений.

С учетом всех изложенных идей и усовершенствований был проведен численный эксперимент, который показал эффективность предложенного метода решения задачи. Динамику изменения значения целевой функции в процессе выполнения процедуры поиска с помощью метода вектора спада можно увидеть на рис. 8.2. Следует заметить, что поиск в большинстве случаев происходил в окрестности радиусов 1 и 2, т. е. за пренебрежимо малое время, поиск в окрестности радиуса 3 тоже происходил быстро, но редко был результативным, а эксперименты с окрестностью радиуса 4 показали, что значительное снижение скорости работы алгоритма не компенсируется нахождением дополнительных минимумов, поэтому было принято решение отказаться от увеличения максимального радиуса.

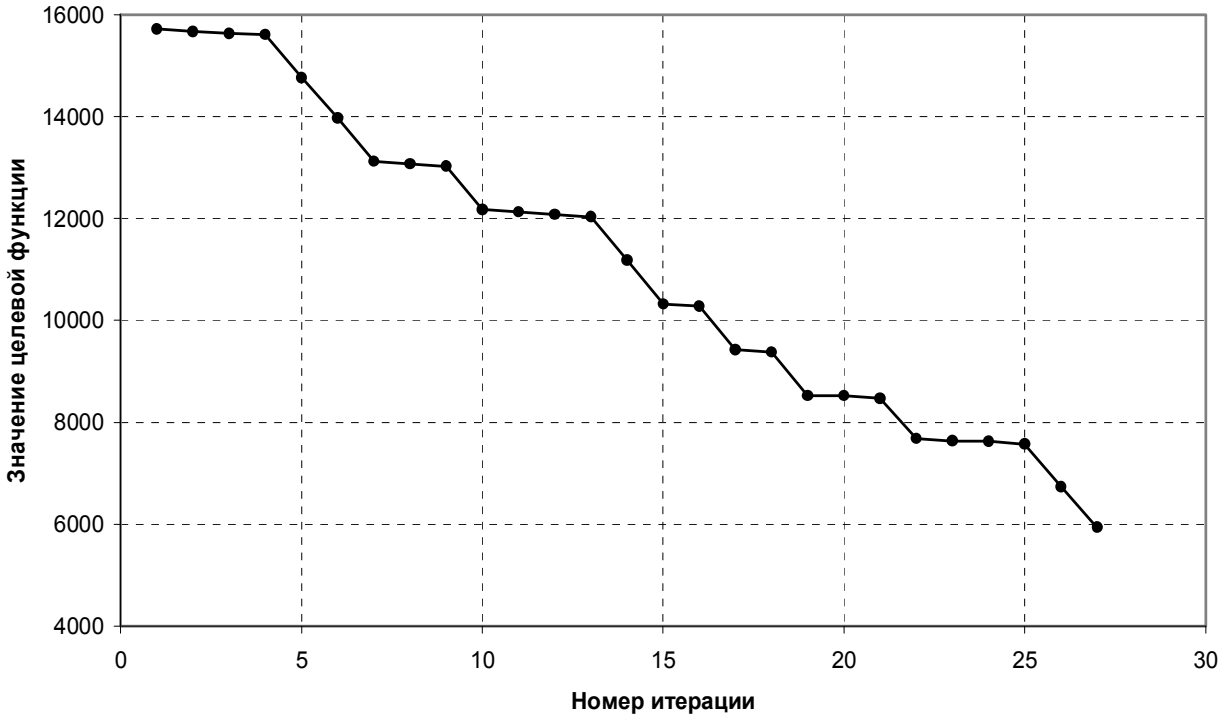


Рис. 8.2. Изменение критерия J в процессе работы алгоритма вектора спада

Также для приведенного примера была исследована сходимость решения к локальному минимуму. Одна из зависимостей, которая иллюстрирует этот процесс, приведена на рис. 8.3. Затенение механизма означает то, что он меняет свое положение на следующей итерации. Соответствующие уровни стека протоколов TCP/IP обозначены римскими цифрами I–IV, а уровень V является фиктивным, и на нем располагаются механизмы, которые не реализуются ни на одном реальном уровне.

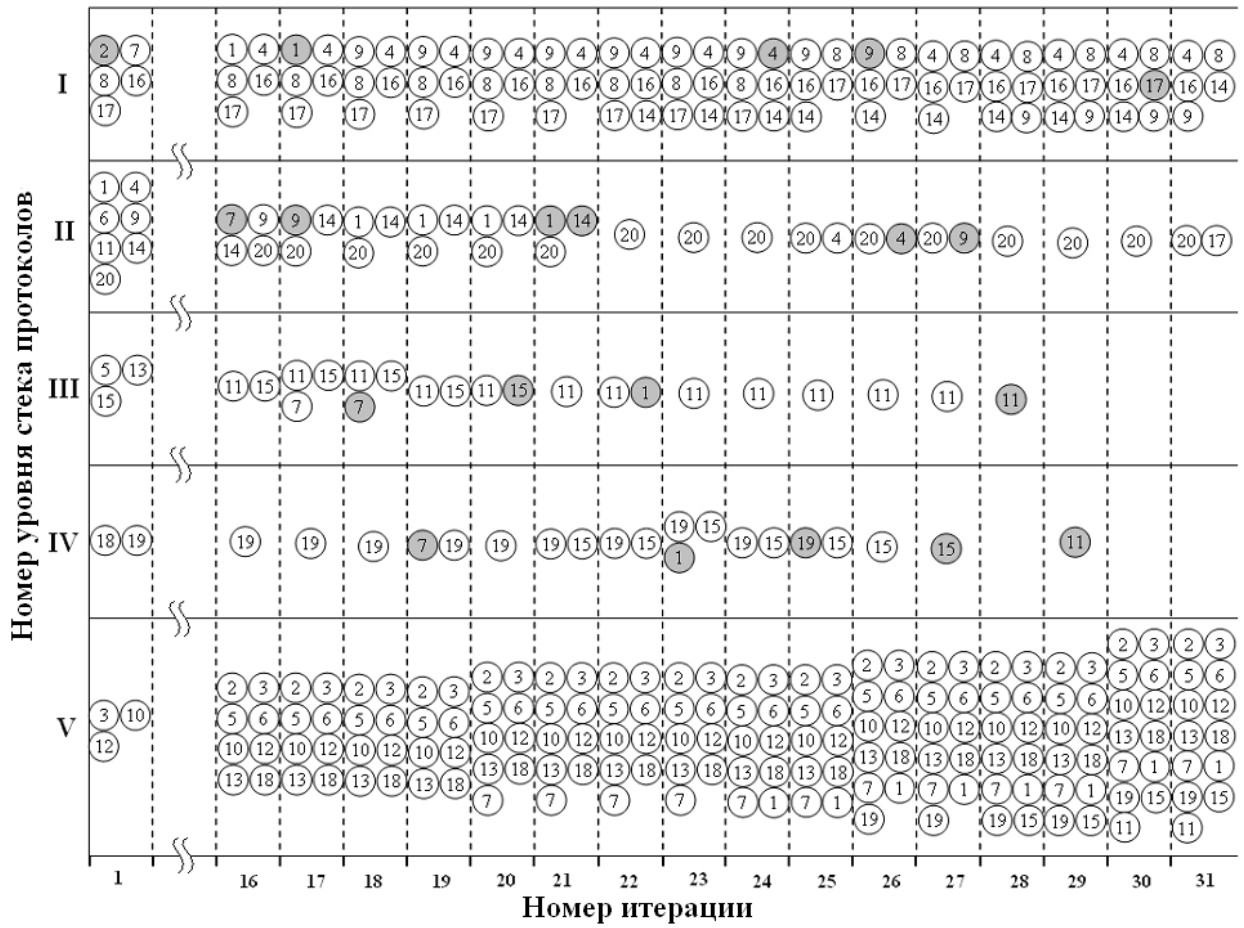


Рис. 8.3. Распределение механизмов защиты по уровням стека TCP/IP на каждой итерации

8.5. Применение алгоритма вектора спада для решения задачи линейного булевого программирования

С целью уменьшения количества переменных задача (8.4) была заменена на эквивалентную задачу сепарабельного программирования, которая была решена с помощью метода вектора спада. В этом пункте используем альтернативный подход, в котором задача линейного булевого программирования (ЛБП) решается прямо с использованием метода вектора спада. Было сделано предположение, что переход к задаче сепарабельного программирования и соответствующее уменьшение ее размерности позволят уменьшить сложность задачи математического программирования. Как показали дальнейшие исследования, при определенной организации вычислений при решении задачи ЛБП ее асимптотическая сложность является такой же, поэтому преимущества перехода к задаче сепарабельного программирования являются

неочевидными. При изменении организации вычислений преимущества может иметь тот или иной вариант. Если существует четкая граница, при превышении которой один вариант преобладает над другим, то на практике целесообразно делать переключение на соответствующий алгоритм, если это не так, то алгоритмы являются несравнимыми. Введем новые обозначения [167]:

$$c_{(i-1)N+j} = C_{ij},\ x_{(i-1)N+j} = M_{ij},\ J_i = \sum_{j=1}^{N} j x_{(i-1)N+j};$$

$$b = \sum_{i=1}^{L}\sum_{j=1}^{N} R_{ij}Q_{ij} - R_{\max},\quad a_{(i-1)N+j} = \sum_{k=j}^{N} R_{ik}Q_{ik}\prod_{k=1}^{j-1}\left(1-R_{ik}\right).$$

Тогда исходная задача может быть записана в виде

$$\min\left\{J(x) = cx \mid ax \le b,\ \sum_{j=1}^{N} x_{(i-1)N+j} \le 1,\ x_i \in \{0,1\},\ i = 1,\ldots,LN\right\}, \qquad (8.7)$$

где a, c, x – векторы размерностей LN.

Условие

$$\sum\nolimits_{j=1}^{N} x_{(i-1)N+j} \le 1 \text{ для } 1 \le i \le L \qquad (8.8)$$

отражает тот факт, что реализация каждого средства безопасности имеет смысл только на одном уровне стека.

Задача (8.7) может быть эффективно решена с использованием метода вектора спада, что относится к классу методов локальной оптимизации. После учета особенностей задачи (8.7) общая схема алгоритма была изменена, поэтому она приводится ниже. Поиск локального минимума происходил в окрестности радиуса 2. В качестве метрики использовалась следующая: $\rho(x, y) = \sum_i |x_i - y_i|$. Обозначим $B_2(x) = \{x' \mid \rho(x, x') \le 2\}$. В результате работы алгоритма находился локальный минимум x'', для которого справедливо $J(x'') \le J(x')$ для всех $x' \in B_2(x'')$.

Алгоритм поиска локального минимума задачи (8.7):

1. Случайно или иным образом выбрать допустимую начальную точку $x = (x_1, \ldots, x_{LN})$.

2. Если в окрестности $B_1(x)$ найдена такая допустимая точка x', что $J(x') < J(x)$, то присвоить $x = x'$ и перейти к п. 3.

3. Если в множестве $B_2(x) \setminus B_1(x)$ найдена такая допустимая точка x', что $J(x') < J(x)$, то присвоить $x = x'$ и перейти к п. 2, иначе перейти к п. 4.

4. Вернуть x – локальный минимум относительно $B_2(x)$.

Основным отличием изложенного алгоритма является то, что ограничения на структуру решения можно эффективно использовать для того, чтобы выполнять некоторые шаги общего метода более рационально. Векторы a, c имеют такую особенность, что их координаты $a_{1+(i-1)N}, a_{2+(i-1)N}, \ldots, a_{N+(i-1)N}$ и $c_{1+(i-1)N}, c_{2+(i-1)N}, \ldots, c_{N+(i-1)N}$ монотонно уменьшаются. Это отражает тот факт, что на нижних уровнях стоимость средств защиты является большей, но и их возможности по уменьшению уровня риска также выше. Другие случаи являются аномальными, поэтому оптимальное направление движения является более прозрачным, чем в общем случае.

Следует отметить, что на шаге 3 алгоритма следует перебирать только те точки, которые находятся на расстоянии $\rho = 2$ от центра, потому что из точек с $\rho = 1$ на предыдущем шаге ни одна не имела лучшего значения целевой функции, чем в текущем центре. Также в любой окрестности не искалась лучшая точка, а сразу после нахождения первой допустимой точки, которая улучшала значение целевой функции, происходила смена текущего центра согласно [150].

Численные эксперименты показали, что увеличение радиуса поиска с $r = 2$ является нецелесообразным, поскольку дополнительное время для поиска в большинстве случаев тратится впустую и новый минимум не находится. Более полезным в смысле получения других минимумов является запуск алгоритма с других начальных точек, и благодаря высокой скорости работы алгоритма количество таких повторных запусков может быть достаточно большим.

Вспомогательные алгоритмы для поиска лучших точек, находящиеся от определенного центра на расстоянии 1 и 2, приведены ниже.

Вспомогательный алгоритм поиска лучшей точки в окрестности радиуса 1:

1. Для текущего центра $(x_1, \ldots, x_{LN})$ вычислить $I = \sum_{i=1}^{LN} a_i x_i$, $T = \sum_{i=1}^{LN} c_i x_i$. Пускай $S = \{1, \ldots, L\}$.

2. Исключить из множества S некий элемент i. Если $J_i = 0$, то перейти к п. 3.

3. Если $I - a_{(i-1)N+J_i} \geq b$, то положить $J_i = 0$, $x_{(i-1)N+J_i} = 0$, $I = I - a_{(i-1)N+J_i}$, $T = T - c_{(i-1)N+J_i}$. Если S пустое, то перейти к п. 4, иначе вернуться к п. 2.

4. Вернуть точку $(x_1, \ldots, x_{LN})$ и значение критерия $T = J(x)$.

Вспомогательный алгоритм поиска лучшей точки в окрестности радиуса 2:

1. Для текущего центра $(x_1, \ldots, x_{LN})$ вычислить $I = \sum_{i=1}^{LN} a_i x_i$, $T = \sum_{i=1}^{LN} c_i x_i$. Пускай $S = \{1, \ldots, L\}$.

2. Исключить из множества S некий элемент i. Если $J_i = 0$, то перейти к п. 3.

3. Если для какого-то $z : J_i < z \leq N$ выполняется $I - a_{(i-1)N+J_i} + a_{(i-1)N+z} \geq b$, то положить $J_i = z$, $x_{(i-1)N+J_i} = 0$, $x_{(i-1)N+z} = 1$, $I = I - a_{(i-1)N+J_i} + a_{(i-1)N+z}$, $T = T - c_{(i-1)N+J_i} + c_{(i-1)N+z}$ и перейти к п. 11.

4. Если S пустое, то перейти к п. 5, иначе вернуться к п. 2.

5. Пускай $S = \{1, \ldots, L\}$, $K = \{1, \ldots, L\}$.

6. Исключить из S индекс i или перейти к п. 11, если S пустое.

7. Исключить из K индекс $j > i$ или перейти к п. 6, если это невозможно.

8. Если $J_i > 1$, а $J_j = 0$, то перейти к п. 10. Если $J_i = 0$, а $J_j > 1$, то поменять местами индексы i и j и перейти к п. 10.

9. Перейти к п. 7.

10. Для всех $0 < z \leq N$. Если $I - a_{(j-1)N+J_j} + a_{(i-1)N+z} \geq b$, то положить $J_i = 0$, $J_j = z$, $x_{(i-1)N+J_i} = 0$, $x_{(j-1)N+z} = 1$, $I = I - a_{(i-1)N+J_i} + a_{(j-1)N+z}$, $T = T - c_{(i-1)N+J_i} + c_{(j-1)N+z}$ и перейти к п. 11.

11. Вернуть точку $(x_1, \ldots, x_{LN})$ и значение критерия T.

Для шага 2 вспомогательного алгоритма поиска лучшей точки в окрестности радиуса 1 и шагов 2 и 6 вспомогательного алгоритма поиска лучшей точки на расстоянии 2 выбор элемента из множества может быть как

последовательным, так и произвольным. При случайном переборе даже при использовании фиксированной начальной точки найденный локальный минимум может отличаться после каждого запуска алгоритма.

Для применения метода вектора спада также необходимо определить некоторое начальное допустимое решение задачи, чтобы потом можно было находить в его окрестности локальный минимум. Одним из вариантов может быть использование наиболее дорогой реализации системы (т. е. все механизмы реализуются на низком уровне). Другой вариант – использовать случайную конфигурацию с корректировкой ее до того состояния, пока она станет допустимой. Соответствующий алгоритм приведен ниже.

Вспомогательный алгоритм выбора начальной точки:

1. Положить $x_k = 0,\ k = \overline{1, LN}$. Пускай $K = \{1, \ldots, L\}$.

2. Для всех $i = \overline{1, N}$ случайным образом выбрать z из интервала $[0, N]$, положить $J_i = z$, а также $x_{(i-1)N+z} = 1$ для случая $z > 0$. Положить $I = \sum_{i=1}^{LN} a_i x_i,\ T = \sum_{i=1}^{LN} c_i x_i$.

3. Пока $I < b$ случайным образом выбрать k из множества K. Иначе перейти к п. 7.

4. Если $J_k = 1$, то исключить k из множества K и вернуться к п. 3.

5. Если $J_k = 0$, то положить $J_k = N,\ x_{kN} = 1,\ I = I + a_{kN},\ T = T + c_{kN}$ и вернуться к п. 3.

6. Положить $J_k = J_k - 1,\ x_{(k-1)N+J_k} = 0,\ x_{(k-1)N+J_k-1} = 1$. Положить $I = I - a_{(k-1)N+J_k} + a_{(k-1)N+J_k-1},\ T = T - c_{(k-1)N+J_k} + c_{(k-1)N+J_k-1}$. Вернуться к п. 3.

7. Вернуть точку $(x_1, \ldots, x_{LN})$.

8.6. Пример синтеза кибернетической системы защиты на основе алгоритма вектора спада

При проведении вычислительного эксперимента параметры задачи генерировались случайным образом. Рассматривался четырехуровневый стек протоколов (например, TCP/IP), т. е. $N = 4$. Количество угроз безопасности равнялось $L = 20$. Движение основного алгоритма начиналось с $V = 100$

начальных точек, а затем среди полученных локальных минимумов (их количество в общем случае будет меньше чем V, поскольку из нескольких начальных точек можно попасть в тот же минимум) выбиралось решение с наименьшим значением целевой функции.

Численные эксперименты показали, что разработанный алгоритм является работоспособным и эффективным и может использоваться как самостоятельно, так и в сочетании с точными методами (как указано в [150], после нахождения допустимого решения с помощью *P*-метода дальше можно уточнять его с помощью метода вектора спада).

Следует отметить, что из-за высокой скорости работы алгоритма его общее время работы, включая многократные запуски из разных начальных точек, будет достаточно малым, при этом можно находить не единственный минимум, а, скажем, K конфигураций с наименьшей стоимостью. Поскольку исходные данные задачи недостаточно точно определены (это касается числовых характеристик кибератак, поскольку стоимости механизмов защиты являются точно известными), то возможно разработать такую процедуру, которая будет статистически оценивать значимость каждого из механизмов защиты при нечетко заданных коэффициентах их эффективности.

Оценка вычислительной сложности алгоритма

Найдем теоретическую зависимость времени работы алгоритма от размерности задачи (количества киберугроз). Очевидно, что время работы линейно зависит от количества точек, которые перебираются при работе алгоритма. Оценим ее, для этого рассмотрим окрестность с радиусом 1 некоторой точки $(x_1,\ldots,x_{LN})$ и оценим количество точек, которые необходимо проверить, чтобы найти среди них лучшую точку. Из всех возможных направлений изменения вектора $(x_1,\ldots,x_{LN})$ большую часть не рассматриваем, так как они являются недопустимыми или из-за нарушения ограничения на величину риска, или из-за нарушения условия (8.6), или из-за того, что при таком изменении увеличивается значение целевой функции. Рассмотрим множество индексов

$$I = \{i \mid J_i > 0,\ i = 1,\ldots,L\},$$

а также будем использовать обозначения $|I|$ для количества элементов множества I. Тогда в этом множестве окажутся номера тех средств безопасности, которые на данный момент используются в КС КЗИ, и можно утверждать,

что для $s \notin I$ изменение значения булевых переменных $x_{(s-1)L+j}$, $j = 1,\ldots,N$, возможно только в направлении увеличения (т. е. с 0 до 1), а для $s \in I$ изменить можно только булевую переменную $x_{(s-1)L+J_s}$, причем в направлении уменьшения. Первый случай в результате положительности коэффициентов c_i приведет к тому, что значение целевой функции будет увеличиваться, т. е. движение будет происходить в неверном направлении. Очевидно, что из всех возможных $2LN$ направлений изменений вектора $(x_1,\ldots,x_{LN})$ двигаться можно только в $|I|$ направлениях. Поскольку $|I| \le L$ (количество средств безопасности, которые использованы в КС КЗИ, не превышает количества имеющихся средств), то L является верхней границей для количества точек, которые будут перебираться в окрестности $B_1(x)$ при поиске лучшей допустимой точки.

Теперь рассмотрим множество, состоящее из точек, удаленных от некоторой точки $(x_1,\ldots,x_{LN})$ на расстояние $\rho = 2$, и оценим количество точек, которые необходимо проверить, чтобы попытаться найти среди них лучшую точку, чем x. Пускай множество индексов I определяется так же, как и раньше, тогда можно утверждать, что для $s \notin I$, $k \in I$ возможны только такие направления движения:

1. Изменяется уровень, на котором используется k-е средство защиты (конкретно, уровень увеличивается, так как при уменьшении растет стоимость и происходит движение в неоптимальном направлении). Для этого необходимо положить $x_{(k-1)L+J_k} = 0$, а потом поочередно присваивать $x_{(k-1)L+J_k+v} = 1$, где $0 < v \le N - J_k$.

2. Вместо k-го средства защиты, которое выводится из употребления, применяется s-е. Для этого необходимо положить $x_{(k-1)L+J_k} = 0$, а потом поочередно присваивать $x_{(s-1)L+v} = 1$, где $v = \overline{N,1}$ (начинать следует с более дешевых верхних уровней, этим объясняется обратный отсчет).

Для первого случая максимальное количество направлений движения составляет $|I|$, а для второго – $|I|(L-|I|)$. Поскольку выше отмечалось, что $|I| \le L$, то общая сумма $|I| + |I|(L-|I|)$ не превышает $(L+1)^2/4$.

Проверим полученную теоретическую зависимость на практике путем проведения численного эксперимента, в котором для каждой размерности задачи будет найдено реальное количество точек, которые перебирались в процессе работы алгоритма. Полученная зависимость количества точек от размерности задачи (количества киберугроз) изображена на рис. 8.4.

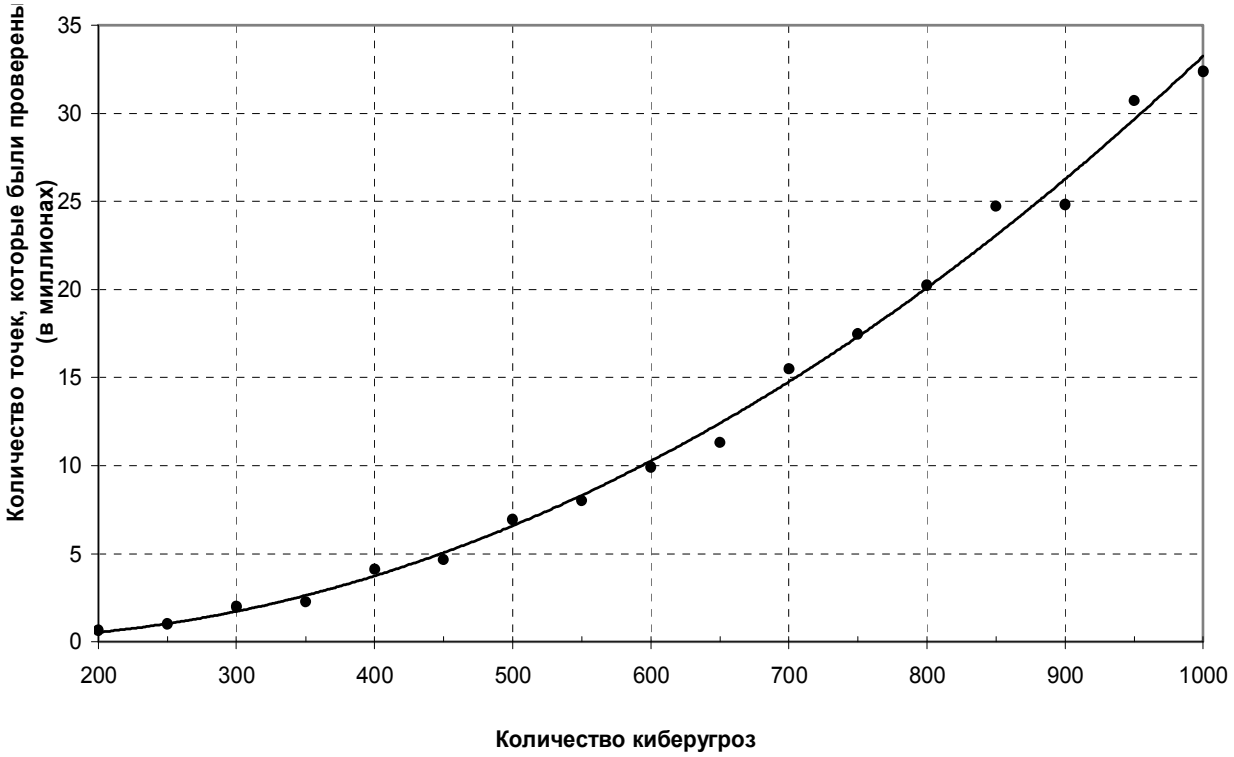


Рис. 8.4. Зависимость количества точек от количества киберугроз

При проведении эксперимента параметры задачи генерировались случайным образом, причем для каждой размерности количество точек усреднялось для десяти запусков. При каждом запуске движение начиналось со 100 начальных точек, а в конце выбиралось решение с наименьшим значением целевой функции.

Можно сделать вывод, что полученная квадратичная зависимость согласуется с теорией, а полиномиальный рост сложности позволяет решать задачи большой размерности.

8.7. Синтез кибернетической системы защиты информации с учетом эффективности механизмов защиты

В п. 8.2 была предложена задача математического программирования для синтеза КС КЗИ с минимальной стоимостью при условии ограничения на величину остаточного риска. Обобщим упомянутую задачу, введя к рассмотрению коэффициент эффективности α_{ij} i-го механизма защиты на j-м уровне стека протоколов. Будем считать, что $0 \le \alpha_{ij} \le 1$. Тогда вероятность того, что i-й механизм защиты нейтрализует киберугрозу на j-м уровне стека, будет равняться $\alpha_{ij} M_{ij}$, где булевая переменная M_{ij} определяет, реализуется ли i-й механизм защиты на j-м уровне стека. После введения коэффициентов эффективности задача математического программирования будет иметь следующий вид [168]:

$$\begin{cases} \sum_{i=1}^{L}\sum_{j=1}^{N} R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\prod_{k=1}^{j}(1-\alpha_{ik}M_{ik})\left(Q_{ij}+\sum_{k=j+1}^{N}R_{ik}Q_{ik}\right) \le R_{\max} \\ \sum_{i=1}^{L}\sum_{j=1}^{N} M_{ij}C_{ij} \to \min\limits_{M_{ij}\in\{0,1\}} \end{cases}, \qquad (8.9)$$

где R_{ij} – вероятность реализации i-й киберугрозы на j-м уровне стека;

Q_{ij} – величина убытков при реализации i-й киберугрозы на j-м уровне стека;

$R_{\max}$ – верхняя граница потерь от нарушения кибербезопасности при реализации L киберугроз;

C_{ij} – стоимость реализации i-го механизма защиты на j-м уровне стека.

Эквивалентная задача сепарабельного программирования

В общем случае задачу (8.9) можно свести к эквивалентной задаче сепарабельного программирования. Для начала будем кодировать комбинации механизмов, нейтрализующих одну и ту же i-ю киберугрозу, в виде целого числа:

$$x_i = \sum_{j=1}^{N} 2^{j-1} M_{ij} \in \{0,\ldots,2^N - 1\},$$

а также будем считать, что значение переменной $x_i = 2^N$ обозначает отказ от реализации i-го средства защиты.

Тогда выражения для стоимости системы и средних потерь от нарушения кибербезопасности можно записать в виде $\sum_{i=1}^{L} F_i(x_i)$ и $\sum_{i=1}^{L} G_i(x_i)$ соответственно. Функции F_i и G_i задаются таблично и зависят от целочисленной переменной x_i неявным образом. Для их вычисления необходимо декодировать значение переменной x_i и подставить полученные булевые значение в соответствующие выражения задачи (8.9) для фиксированного i:

$$\begin{cases} F_i(x_i) = \begin{cases} \sum_{j=1}^{N} M_{ij} C_{ij}, & x_i = 0,\ldots,2^N - 1, \\ 0, & x_i = 2^N, \end{cases} \\ G_i(x_i) = \begin{cases} \sum_{j=1}^{N} R_{ij} \prod_{k=1}^{j-1} (1 - R_{ik}) \prod_{k=1}^{j} (1 - \alpha_{ik} M_{ik}) \left(Q_{ij} + \sum_{k=j+1}^{N} R_{ik} Q_{ik} \right), & x_i = 0,\ldots,2^N - 1, \\ \sum_{j=1}^{N} R_{ij} Q_{ij}, & x_i = 2^N. \end{cases} \end{cases}$$

Задача в этом случае имеет вид

$$\min_{x_i \in \{0,\ldots,2^N\}} \left\{ \sum_{i=1}^{L} F_i(x_i) \,\middle|\, \sum_{i=1}^{L} G_i(x_i) \le R_{\max} \right\} \tag{8.10}$$

и может решаться с помощью метода ПАВ [149] или с использованием метода вектора спада для задачи сепарабельного программирования, описанного в предыдущих пунктах.

Следует отметить, что также легко рассмотреть случаи, когда некоторые комбинации переменных M_{ij} для фиксированного i запрещены (например, когда средство не может быть реализовано на определенных уровнях стека протоколов). В этом случае из всех возможных 2^N значений x_i некоторые нужно отбросить, а остальные занумеровать заново. Полученная задача имеет вид

$$\min_{x_i \in X_i} \left\{ \sum_{i=1}^{L} F_i'(x_i) \,\middle|\, \sum_{i=1}^{L} G_i'(x_i) \le R_{\max},\ X_i = \{0,\ldots,l_i\},\ l_i < 2^N \right\},$$

где функции F_i' и G_i' задаются заново для всех значений $x_i = 0,\ldots,l_i$, которые остались.

Кроме рассмотренного выше общего случая, когда мы не делаем никаких предположений относительно значений коэффициентов прочности средств

защиты и возможных комбинаций этих средств, реализованных на различных уровнях, существуют ситуации, когда задача (8.9) становится линейной. Рассмотрим их.

Линеаризация задачи

Рассмотрим ситуацию, когда для нейтрализации киберугрозы избирается реализация средства защиты на одном уровне, или мы вообще отказываемся от его использования. Для случая абсолютно надежных средств защиты ($\alpha_{ij} = 1$) только среди таких конфигураций КС КЗИ следует искать оптимальную, а все те, в которых реализация средства происходит на двух и более уровнях, имеют большую стоимость. В этом случае задача (8.9) на самом деле будет линейной. С введением неединичных коэффициентов эффективности средств защиты их дублирование на разных уровнях стека протоколов уже будет оправданным, и для приведения задачи (8.9) к линейному виду необходимо принудительно наложить ограничения на структуру КС КЗИ. В этом случае для i-й киберугрозы справедливо $\sum_{j=1}^{n} M_{ij} \le 1$, $1 \le n \le N$, а вместе с тем и $\prod_{j=1}^{n}(1-\alpha_{ij}M_{ij}) = 1 - \sum_{j=1}^{n}\alpha_{ij}M_{ij}$, ибо все перекрестные произведения исчезают. Итак, в этом случае ограничение на значения средних потерь будет линейным:

$$\sum_{i=1}^{L}\sum_{j=1}^{N} R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\sum_{k=1}^{j}\alpha_{ik}M_{ik}\left(Q_{ij} + \sum_{k=j+1}^{N} R_{ik}Q_{ik}\right) \ge$$

$$\ge \sum_{i=1}^{L}\sum_{j=1}^{N} R_{ij}\prod_{k=1}^{j-1}(1-R_{ik})\left(Q_{ij} + \sum_{k=j+1}^{N} R_{ik}Q_{ik}\right) - R_{\max}.$$

После наложения дополнительного ограничения на структуру решения и тождественных преобразований исходную задачу можно записать в линейном виде:

$$\begin{cases} \sum_{i=1}^{L}\sum_{j=1}^{N} M_{ij}C_{ij} \to \min\limits_{M_{ij}\in\{0,1\}}, \ \sum_{j=1}^{N} M_{ij} \le 1,\ i = \overline{1,L} \\ \sum_{i=1}^{L}\sum_{j=1}^{N} M_{ij}\left(\alpha_{ij}\sum_{k=j}^{N} R_{ik}Q_{ik}\prod_{k=1}^{j-1}(1-R_{ik})\right) \ge \sum_{i=1}^{L}\sum_{j=1}^{N} R_{ij}Q_{ij} - R_{\max} \end{cases}. \qquad (8.11)$$

При всех $\alpha_{ij} = 1$ (т. е. в случае абсолютно надежных средств защиты) задача (8.11) сводится к задаче линейного булевого программирования (ЛБП) из предыдущего пункта.

Задачу (8.11) можно решать с помощью точных методов ЛБП (метода Балаша с усовершенствованием Джеофриона [160], метода ПАВ, *P*-метода [158] и др.) и методов локальной оптимизации (метода вектора спада, локально-стохастических алгоритмов и метода управляемого случайного поиска [150; 161]).

Если задача (8.11) не решается прямо, а происходит переход к эквивалентной задаче сепарабельного программирования, то функции F_i и G_i из (8.10) выражаются следующим образом:

$$F_i(x_i) = \begin{cases} \sum_{j=1}^{N} M_{ij} C_{ij},\ x_i = 0, \ldots, N-1, \\ 0,\ x_i = N, \end{cases}$$

$$G_i(x_i) = \begin{cases} \sum_{j=1}^{N} R_{ij} Q_{ij} - \alpha_{i,x_i+1} \sum_{k=x_i+1}^{N} R_{ik} Q_{ik} \prod_{k=1}^{x_i} (1 - R_{ik}),\ \ x_i = 0, \ldots, N-1, \\ \sum_{j=1}^{N} R_{ij} Q_{ij},\ x_i = N. \end{cases}$$

Решение задачи сепарабельного программирования с помощью локально-стохастических алгоритмов

В предыдущем пункте для решения задачи (8.10) было предложено использовать метод вектора спада, который относится к классу методов локальной оптимизации. Кроме оригинального метода вектора спада, также существует ряд других локально-стохастических методов [161], поэтому предлагается использовать их идеи для дальнейшего совершенствования алгоритма решения задачи. Важную роль при работе локально-стохастических алгоритмов играет процедура генерирования случайной перестановки, которая приводится ниже.

Алгоритм генерации случайной перестановки Π_α

1. Формируем начальную последовательность индексов $J_0 = (j_1^0, \ldots, j_n^0)$ и задаем некоторое значение целочисленного параметра α.

2. Предполагаем $k = n$.

3. С помощью генератора псевдослучайных чисел, равномерно распределенных на интервале $(0, 1)$, получаем случайное число ξ и вычисляем $i = [(1 - \xi^\alpha)k + 1]$.

4. Меняем местами i-й и k-й элементы последовательности J_{n-k}, получая в результате последовательность J_{n-k+1}.

5. Заменяем k на $k-1$ и при $k>1$ переходим к п. 3, а при $k=1$ возвращаем случайную перестановку $J=J_{n-1}$.

Если рассматривать метод вектора спада с радиусом окрестности $r=1$, то всем переменным можно придать некоторый приоритет, который будет отражать их «перспективность» с точки зрения движения в направлении их роста. Эвристическое правило для вычисления таких приоритетных коэффициентов для j-й переменной может выглядеть следующим образом:

$$v_j=(F_j(x_j^0)-F_j(x_j^0+1))\big/(G_j(x_j^0+1)-G_j(x_j^0)).$$

Для случая $x_j^0=N$ дальнейшее движение невозможно, поэтому эта переменная не учитывается при вычислении приоритетных коэффициентов. Случай $G_j(x_j^0+1)=G_j(x_j^0)$ нетипичен, потому что обычно механизм, реализованный на более высоком уровне стека, обеспечивает меньший уровень защиты.

Процедура Π_α имеет целью сделать перебор направлений изменения координат случайным, не меняя существенно порядок, который определяется с помощью приоритетных коэффициентов для соответствующих переменных. С увеличением параметра α полученная перестановка J приближается к J_0, а при $\alpha=1$ приоритет не учитывается вообще и перестановка становится случайной. С учетом особенностей задачи алгоритм циклического координатного подъема будет иметь следующий вид.

Алгоритм циклического координатного подъема

1. В качестве начального приближения x^0 выбираем или нулевой вектор, или другое допустимое решение задачи. Полагаем $R=0$ и фиксируем некоторое значение параметра α.

2. Вычисляем по формуле $v_j=(F_j(x_j^0)-F_j(x_j^0+1))\big/(G_j(x_j^0+1)-G_j(x_j^0))$ приоритетные коэффициенты v_j, $j=1,\ldots,L$ и строим такую адаптационную последовательность $J_0=(j_1^0,\ldots,j_L^0)$, что $v_{j_1^0}\geq\ldots\geq v_{j_L^0}$. Исключаем из J_0 все индексы, которые соответствуют $x_i^0=N$ (максимальному значению переменной). Пусть количество индексов после этой процедуры сократилась от L до K.

3. С помощью процедуры Π_α генерируем случайную перестановку $J = (j_1, \ldots, j_K)$, $K \le L$ элементов последовательности J_0.

4. Выходим из точки x^0 и последовательно в порядке $j_1, \ldots, j_K$ определяем координаты допустимой точки из окрестности радиуса $r = 1$, что улучшает значение целевой функции.

5. Полагаем $h = 1$ и $x = x^0$.

6. Если $R - G_{j_h}(x_{j_h}) + G_{j_h}(x_{j_h} + 1) \le R_{\max}$, то полагаем $R = R - G_{j_h}(x_{j_h}) + G_{j_h}(x_{j_h} + 1)$, $x_{j_h} = x_{j_h} + 1$.

7. Если $h = K$, то полученная точка является точкой локального минимума, и переходим к п. 10. Иначе переходим к п. 9.

8. Если $x \ne x^0$, то полагаем $x^0 = x$ и переходим к п. 2. Иначе полагаем $x^* = x$ и переходим к п. 10.

9. Есть $h < K$. Заменяем h на $h + 1$ и переходим к п. 6.

10. Возвращаем точку локального минимума x^* целевой функции относительно окрестности радиуса $r = 1$.

В [161] также был предложен алгоритм бикоординатного подъема, который является локально-стохастическим алгоритмом поиска минимума в окрестности с радиусом $r = 2$. Для рассматриваемой задачи алгоритм бикоординатного подъема можно изложить в следующем виде.

Алгоритм бикоординатного подъема

1. С помощью алгоритма циклического координатного подъема находим точку x минимума целевой функции относительно окрестности единичного радиуса. Вычисляем значение $R = \sum_{i=0}^{L} G_i(x_i)$.

2. Рассмотрим два множества U и V, определяемые следующим образом: $U = \{i \mid x_i < N\}$, $V = \{i \mid x_i > 0\}$. Обозначим $|U| = N_U$ и $|V| = N_V$.

3. Формируем множество Z, состоящее из троек $\{z_i, f_i, q_i\}$, где $f_i = F_{z_i}(x_{z_i}) - F_{z_i}(x_{z_i+1})$, $q_i = '+'$ для $z_i \in U$ и $f_i = F_{z_i}(x_{z_i-1}) - F_{z_i}(x_{z_i})$, $q_i = '-'$ для $z_i \in V$. Упорядочиваем элементы множества Z по возрастанию f_i.

4. Полагаем $\bar{x} = x$.

5. Полагаем $l = \min\{j \mid q_j = '-'\}$, $k = \max\{j \mid q_j = '+'\}$.

6. Если $k < l$, то x является точкой минимума целевой функции относительно окрестности радиуса 2 и работа прекращается.

7. Есть $k > l$. Если $z_k = z_l$, то переходим к п. 9.

8. Если выполняется $R + G_{z_k}(x_{z_k} + 1) - G_{z_k}(x_{z_k}) + G_{z_l}(x_{z_l} - 1) - G_{z_l}(x_{z_l}) < R_{\max}$, то полагаем $x_{z_k} = x_{z_k} + 1$, $x_{z_l} = x_{z_l} - 1$ и переходим к п. 2. Иначе переходим к следующему пункту.

9. Если $W_U = \{j \mid q_j = '+',\ j < k\} \neq \varnothing$, то полагаем $k = \max\limits_{j \in W_U} j$ и переходим к п. 11. Иначе переходим к п. 10.

10. Если $W_V = \{j \mid q_j = '-',\ j > l\} \neq \varnothing$, то полагаем $l = \min\limits_{j \in W_V} j$, $k = \max\{j \mid q_j = '+'\}$ и переходим к п. 11. Иначе переходим к п. 12.

11. Если $k < l$, то переходим к п. 12, иначе – к п. 7.

12. Если $x = \bar{x}$, то точка x является искомой точкой локального минимума относительно окрестности радиуса 2. Иначе с помощью алгоритма циклического координатного подъема, исходя из точки x, как начальной, находим локально-оптимальное решение x'. Если $x' = x$, то x является искомой точкой локального минимума относительно окрестности радиуса 2. Иначе переходим к п. 2, полагаем $x = x'$.

Применение *P*-метода для решения задачи булевого программирования

Изложим алгоритм решения задачи ЛБП (8.11), которая возникает в процессе синтеза оптимальной КС КЗИ, с использованием *P*-метода.

Введем обозначения:

$$b_{L+1} = R_{\max} - \sum_{i=1}^{L}\sum_{j=1}^{N} R_{ij}Q_{ij}, \quad a_{L+1,(i-1)N+j} = -\alpha_{ij}\sum_{k=j}^{N} R_{ik}Q_{ik}\prod_{k=1}^{j-1}(1-R_{ik})$$

$$b_s = 1, \quad a_{sk} = \begin{cases} 1, & (s-1)N < k < sN \\ 0, & \textit{иначе.} \end{cases}, \ s = \overline{1, L},$$

$$c_{(i-1)N+j} = C_{ij}, \quad x_{(i-1)N+j} = M_{ij}.$$

Тогда исходная задача (8.11) может быть записана в виде

$$\min\{J(x) = cx \mid Ax \le b,\ x = (x_1, \ldots, x_{LN}),\ x_j \in \{0, 1\},\ j = 1, \ldots, LN\},$$

где A – матрица $((L+1) \times LN)$;

b, c, x – векторы размерности LN.

Эта задача может быть решена с использованием *P*-метода [150, 158]. Этот метод относится к классу комбинаторных и заключается в том, что все значения вектора неявно перебираются в порядке увеличения в смысле лексикографического упорядочения, при этом бесперспективные частичные решения НЕ достраиваются, что значительно сокращает время поиска.

Для доказательства несовместимости подсистемы $(l+1)$-го порядка для частичного l-решения $(x_1,\ldots,x_l)$ используются как исходные неравенства задачи (8.11), так и одно искусственное, в котором мы отражаем наше знание о значении верхней границы целевой функции, известной на данный момент. До того момента, пока мы не нашли первое допустимое решение, можно положить $J^* = \infty$. Если же мы находим некоторое полное допустимое решение $(x'_1,\ldots,x'_{LN})$, то необходимо сразу положить $J^* = \sum_{j=1}^{LN} c_j x'_j - \delta$, где $\delta > 0$, и для всех допустимых векторов $x \neq y$ выполняется $|J(x) - J(y)| \geq \delta$. Для вычислений с конечной точностью величина δ всегда известна. Так, для целых c_j имеем $\delta = \text{НСД}(c_1,\ldots,c_{LN})$. Для рациональных коэффициентов нетрудно получить аналогичное выражение. В том случае, если мы положим $\delta = 0$, в результате работы алгоритма будет найден лексикографически максимальный вектор из множества оптимальных решений. Поскольку векторы перебираются в порядке их лексикографического роста, то на это тратится дополнительное время, что нежелательно.

Особенности задачи позволяют сделать определенные упрощения. Так, в общем случае в алгоритме используются два множества Z_j и W_j, заполняемые перед началом его работы. Если бы знаки и значения коэффициентов матрицы A не были известны заранее, то мы бы должны были вычислять и сохранять все $2LN \times (L+2)$ элементов множеств Z_j и W_j. На самом деле мы знаем, что в первых L неравенствах все коэффициенты являются булевыми, а часть матрицы A, отвечающая за эти неравенства, имеет ту особенность, что в каждом ее столбце есть только одна единица, а в каждой строке есть ровно N единиц. Для $(L+1)$-го неравенства известно, что все его коэффициенты являются отрицательными. Также известно, что искусственное $(L+2)$-е неравенство, которое используется для отсева по значению целевой функции, имеет все

положительные коэффициенты. Все это приводит к тому, что в нашем случае множества Z_j и W_j имеют следующий вид: для каждого $j = \{1,\ldots,LN\}$ определяем

$$Z_j = \{z_j \mid z_j = \sum_{k=j}^{LN} a_{L+1,k}\},$$

$$W_j = \{w_{ij} \mid w_{ij} = \sum_{k=j}^{LN} a_{ik},\ i = 1,\ldots,L,\ \text{и}\ w_{L+1,j} = \sum_{k=j}^{LN} c_k\}.$$

Как видим, для хранения элементов множества Z_j достаточно LN ячеек, а для хранения элементов множества W_j – $2LN$ ячеек (при этом учтено, что w_{ij} для $i = 1,\ldots,L$ полностью определяется N целочисленными индексами, поэтому из всех $2LN$ ячеек половина всегда будет содержать целые числа).

Среди других особенностей, по которым предложенный алгоритм отличается от оригинального P-метода, является условие, при выполнении которого мы делаем вывод о несовместимости подсистемы $(l+1)$-го порядка для частичного l-решения $(x_1,\ldots,x_l)$. Отличие заключается в том, что некоторые коэффициенты множеств Z_j и W_j известны заранее, поэтому не хранятся, и соответствующие условия должны быть записаны в явном виде.

Пускай I – множество индексов, состоящее из номеров тех ограничений, которые могут быть использованы для доказательства несовместимости подсистемы $(l+1)$-го порядка для частичного l-решения $(x_1,\ldots,x_l)$. Если для подсистемы l-го порядка это множество неизвестно, то можно положить $I = (1,\ldots,L+2)$. В противном же случае, когда мы переходим от подсистемы l-го порядка к подсистеме $(l+1)$-го порядка (очевидно, что мы были вынуждены сделать этот переход из-за того, что попытки доказать несовместимость предыдущей системы были неудачными), из множества I можно изъять некоторые индексы.

Поскольку при выполнении одного из неравенств

$$\begin{gathered} b_{L+1} - \sum_{j=1}^{l} a_{L+1,j} x_j > 0, \\ J^* - \sum_{j=1}^{l} c_j x_j > w_{L+1,l+1}, \\ b_p - \sum_{j=1}^{l} a_{pj} x_j > w_{p,l+1},\ \ p = 1,\ldots,L, \end{gathered} \tag{8.12}$$

для определенного значения $p = 1,\ldots, L+2 \in I$ соответствующее p-е неравенство задачи (в качестве $(L+2)$-го неравенства используем искусственное ограничение на значение целевой функции) никогда не будет нарушаться при увеличении l и произвольном дополнении частичного l-решения, то мы изымаем индекс p из множества I. Таким образом, даже если несовместимость подсистемы $(l+1)$-го порядка нельзя доказать, то при формировании подсистемы $(l+2)$-го порядка в нее можно не включать неравенства с номерами, которые были удалены из множества I. Вместе с тем, неравенства из (8.12), которые нарушаются, могут быть использованы для доказательства несовместимости подсистемы $(l+1)$-го порядка для частичного l-решения $(x_1,\ldots,x_l)$.

Утверждение 1.

Если существует такое p, для которого нарушается одно из неравенств (8.12) и выполняется соответствующее неравенство

$$\sum_{j=1}^{l} a_{p,j} x_j > 1, \qquad p = 1,\ldots, L,$$

$$b_p - \sum_{j=1}^{l} a_{p,j} x_j < z_{l+1}, \quad p = L+1,$$

$$\sum_{j=1}^{l} c_j x_j > J^*, \qquad p = L+2,$$

то подсистема $(l+1)$-го порядка для частичного l-решения $(x_1,\ldots,x_l)$ несовместима.

Для увеличения скорости работы оригинального *P*-метода его авторы предложили использовать эвристическое правило для упорядочения переменных. В нашем случае оно имеет следующий вид:

1. Для всех x_j, $j = 1,\ldots, LN$ находим оценку

$$v_j = \left(\frac{a_{L+1,j}}{b_{L+1}} - 1 \right) c_j,$$

2. Переменные упорядочиваем в порядке возрастания оценок v_j, $j = 1,\ldots, LN$.

Алгоритм *P*-метода

1. Полагаем $t = 1$, $(x_1,\ldots,x_{LN}) = (0,\ldots,0)$, $J^* = \infty$.

2. Если выполняется $R_{\max} \le \sum_{i=1}^{L} \sum_{j=1}^{N} R_{ij} Q_{ij}$, то переходим к п. 3. Иначе задача не имеет решения и работа алгоритма прекращается.

3. Изменяя значение индекса l от t до $LN-1$ и проверяя условия утверждения 1, находим минимальное l', при котором подсистема $(l'+1)$-го порядка для частичного l'-решения $(x_1,\ldots,x_{l'})$ несовместима, и переходим к п. 4. Если же такого l' не существует, то полагаем $x_l = 0,\ l = t,\ldots,LN-1$ и переходим к п. 5.

4. Если все компоненты частичного l'-решения, которое рассматривается, равны 1, то прекращаем работу алгоритма. При этом последнее допустимое решение и будет оптимальным. Иначе строим новое l''-решение по правилам: полагаем $l'' = \max\{j \mid j \in \{1,\ldots,l'\}\ x_j = 0\}$. Полагаем $x_{l''} = 1$, а для случая $l' > l''$ еще и $x_j = 0,\ j = l''+1,\ldots,l'$ и переходим к п. 2.

5. Вычисляем $D_i = b_i - \sum_{j=1}^{LN-1} a_{ij}x_j \quad \forall i = 1,\ldots,L+1,\quad D_{L+2} = J^* - \sum_{j=1}^{LN-1} c_j x_j$.

Если $D_i \geq 0$ для всех $i = 1,\ldots,L+2$, то полагаем $x_{LN} = 0$, получаем допустимое решение $(x_1,\ldots,x_{LN})$ исходной задачи и переходим к п. 6.

6. Полагаем $x_{LN} = 1$. Если $D_i - a_{i,LN} \geq 0 \quad \forall i = 1,\ldots,L+1$ и $D_{L+2} - c_{LN} \geq 0$, то получаем допустимое решение задачи и переходим к п. 7. Иначе полагаем $l' = LN-1$ и переходим к п. 4.

7. После получения первого допустимого решения $(x'_1,\ldots,x'_{LN})$ начинается отсев решений по значению целевой функции. С этой целью полагаем $J^* = \sum_{j=1}^{LN} c_j x_j - \delta$, где δ определено ранее. Полагаем $l' = LN-1$ и переходим к п. 4.

Исследование разработанных алгоритмов и их сравнительная характеристика

Изложенные выше локально-стохастические алгоритмы решения задачи дискретного сепарабельного программирования (8.10) и *P*-метод решения задачи ЛБП (8.11) были запрограммированы на языке C#, и их эффективность исследовалась на модельных примерах с максимальным количеством киберугроз $L = 20$ и количеством уровней стека протоколов $N = 4$. При решении задач ЛБП также делалось сравнение с эффективностью работы алгоритмов Балаша, с модификациями Джеофриона и методом вектора спада, которые

были реализованы на языке Fortran в библиотеке численного анализа НИВЦ МГУ [162] (соответственно, процедуры MNR3R и MLC6R).

Для точных методов Балаша и *P*-метода эффективность оказалась недостаточной, и задачи с количеством булевых переменных $LN = 80$ и $LN = 64$ не были решены за отведенное время. Зато приближенные методы показали высокую эффективность и достаточную для практических применений точность.

Динамику изменения стоимости очередной допустимой конфигурации КС КЗИ в процессе работы *P*-метода можно проследить на рис. 8.5. Временные характеристики упомянутых алгоритмов приведены в табл. 8.1. Время работы метода вектора спада при решении задачи ЛБП и алгоритма бикоординатного подъема при решении задачи сепарабельного программирования является полиномиальным, что проиллюстрировано на рис. 8.6.

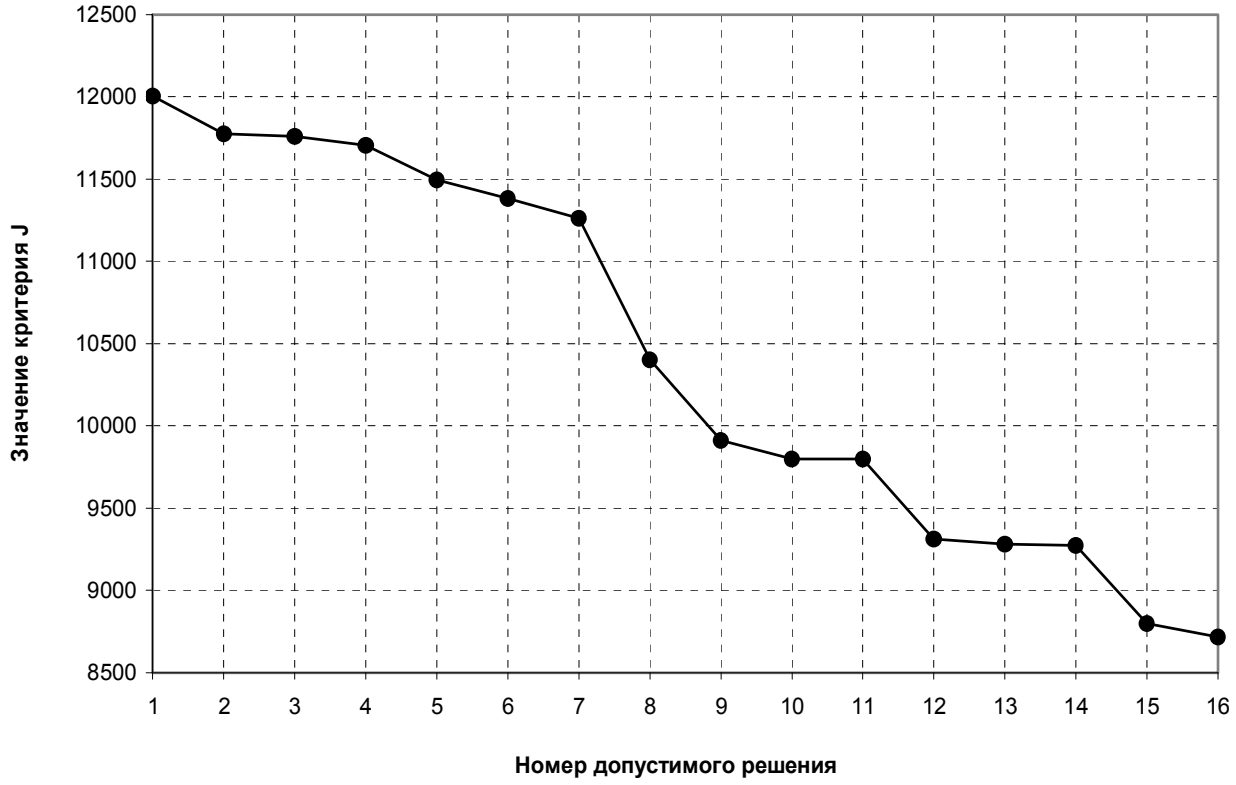


Рис. 8.5. Изменение значения критерия качества в процессе работы *P*-метода

Таблица 8.1

Временные характеристики алгоритмов

Количество угроз L	Время работы алгоритмов			
	MNR3R	**MLC6R**	***P*-метод**	**Бикоорд. подъем**
10	0,0313 с	0,195 мс	0,08 с	0,4995 мс
11	0,1458 с	0,255 мс	0,31 с	0,5715 мс
12	0,3327 с	0,336 мс	0,78 с	0,7309 мс
13	0,5195 с	0,405 мс	3,20 с	0,7985 мс
14	1,526 с	0,508 мс	11,53 с	0,9440 мс
15	32,33 с	0,621 мс	49,25 с	1,1275 мс
16	—	—	2 мин 11 с	—
17	—	—	7 мин 45 с	—
18	—	—	31 мин 19 с	—
19	—	—	1 ч 43 мин 35 с	—
20	—	1,305 мс	—	2,0333 мс

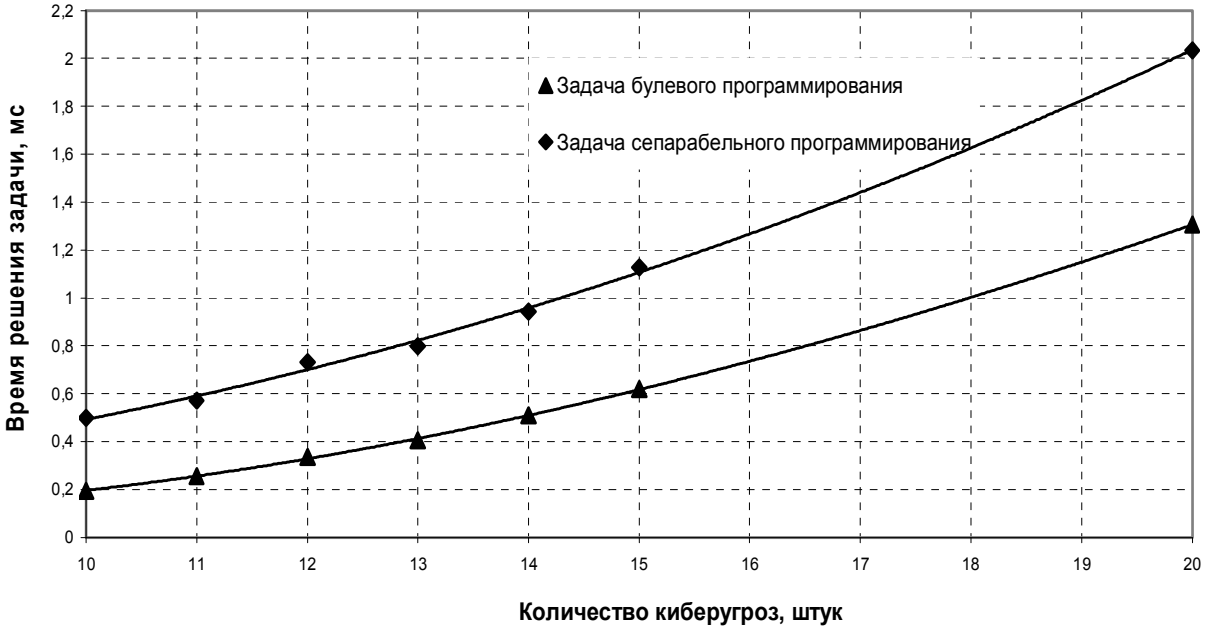


Рис. 8.6. Зависимость времени работы приближенных методов от размерности задачи

8.8. Синтез кибернетической системы защиты информации с максимальной вероятностью сохранения безопасности

Остаточный риск, как интегральный показатель потенциальных потерь, которые могут возникнуть после успешной реализации кибератаки в определенной киберсистеме, в настоящее время рассматривается в качестве одной из характеристик и является критерием качества этой КС КЗИ. В отличие от

подхода концепции «нулевого риска», которая долго господствовала при проектировании структурно сложных систем различной природы, в современном мире все чаще используется концепция управления рисками, которая предполагает существование определенных потерь при реализации угроз в системе, которые не приводят к полному прекращению ее функционирования [98].

В предыдущей главе была предложена математическая модель, описывающая вероятность сохранения защищенности информации в ИКС с открытой архитектурой. Также была сформулирована задача математического программирования, которая возникает при синтезе системы КС КЗИ, которая должна минимизировать вероятность нарушения кибербезопасности при наложении ограничения на стоимость средств защиты, входящих в ее состав. Задача математического программирования выглядела так:

$$\begin{cases} \prod\limits_{i=1}^{L}\left[1-\sum\limits_{j=1}^{N}\left(E_{ij}\prod\limits_{k=1}^{j-1}(1-E_{ik})\prod\limits_{k=1}^{j}\left(1-\alpha_{ik}M_{ik}\right)\right)\right]\to \max\limits_{M_{ik}\in\{0,1\}}, \\ \sum\limits_{i=1}^{L}\sum\limits_{j=1}^{N}M_{ij}C_{ij}\le C_{\max}, \end{cases} \tag{8.13}$$

где N – количество уровней стека протоколов ИКС;

L – количество киберугроз информации и средств защиты;

E_{ik} – показатель эффективности реализации i-й киберугрозы информации на k-м уровне;

M_{ik} – параметр, значение которого (0 или 1) определяет наличие i-го механизма защиты, реализованного на k-м уровне;

α_{ik} – коэффициент эффективности i-го механизма защиты, реализованного на k-м уровне.

Случай с отсутствием дублирования средств на различных уровнях

Поскольку для случая абсолютно надежных средств защиты их реализация имеет смысл только на одном уровне стека протоколов (неоптимальность конфигураций, в которых средства дублируются на нескольких уровнях, показана в предыдущей главе), то общее выражение для целевой функции задачи (8.13) можно упростить. Также подобную конфигурацию КС КЗИ можно создать искусственно. Хотя она и не обязательно будет оптимальной, но может оказаться приемлемой. В обоих случаях для каждой i-й киберугрозы

выполняется $\sum_{j=1}^{N} M_{ij} \le 1$. Это неравенство позволяет утверждать, что выполняется $\prod_{j=1}^{n}(1-\alpha_{ij}M_{ij}) = 1-\sum_{j=1}^{n}\alpha_{ij}M_{ij}$ (для абсолютно надежных средств защиты $\alpha_{ij}=1$, а для общего случая $0 \le \alpha_{ij} \le 1$). Тогда общий вид задачи (8.12) упрощается, так как выполняются равенства:

$$\prod_{i=1}^{L}\left[1-\sum_{j=1}^{N}\left(E_{ij}\prod_{k=1}^{j-1}(1-E_{ik})\prod_{k=1}^{j}(1-\alpha_{ik}M_{ik})\right)\right]=$$

$$=\prod_{i=1}^{L}\left[1-\sum_{j=1}^{N}\left(E_{ij}\left(1-\sum_{k=1}^{j}\alpha_{ik}M_{ik}\right)\prod_{k=1}^{j-1}(1-E_{ik})\right)\right]=$$

$$=\prod_{i=1}^{L}\left[1-\sum_{j=1}^{N}\left(E_{ij}\prod_{k=1}^{j-1}(1-E_{ik})\right)+\sum_{j=1}^{N}\left(E_{ij}\sum_{k=1}^{j}\alpha_{ik}M_{ik}\prod_{k=1}^{j-1}(1-E_{ik})\right)\right],$$

$$\sum_{j=1}^{N}\left(E_{ij}\sum_{k=1}^{j}\alpha_{ik}M_{ik}\prod_{k=1}^{j-1}(1-E_{ik})\right)=\sum_{j=1}^{N}\alpha_{ij}M_{ij}\prod_{k=1}^{j-1}(1-E_{ik})(1-\prod_{k=j}^{N}(1-E_{ik})),$$

$$1-\sum_{j=1}^{N}E_{ij}\prod_{k=1}^{j-1}\left(1-E_{ik}\right)=\prod_{j=1}^{N}\left(1-E_{ij}\right).$$

Обозначим $A_{ij}=\alpha_{ij}\left(\prod_{k=1}^{j-1}\left(1-E_{ik}\right)-\prod_{j=1}^{N}\left(1-E_{ij}\right)\right)$, $B_i=\prod_{j=1}^{N}\left(1-E_{ij}\right)$, тогда в сокращенном виде задача синтеза КС КЗИ, которая обеспечивает максимальную вероятность сохранения безопасности при наложении ограничений на стоимость системы и отсутствия дублирования средств, будет иметь следующий вид:

$$\begin{cases}\prod_{i=1}^{L}\left(B_i+\sum_{j=1}^{N}A_{ij}M_{ij}\right)\to\max\limits_{M_{ij}\in\{0,1\}},\\ \sum_{i=1}^{N}M_{ij}\le 1,\ j=\overline{1,L},\\ \sum_{i=1}^{L}\sum_{j=1}^{N}M_{ij}C_{ij}\le C_{\max}.\end{cases} \tag{8.14}$$

Полученная задача нелинейного булевого программирования может решаться как точными методами, так и приближенными. Для решения задачи (8.14) приближенными методами, в частности локально-стохастическими, запишем ее в виде задачи сепарабельного программирования. Такая запись также позволит применить точный метод ПАВ. Итак, вместо булевых переменных, обозначающих, на каком уровне мы реализуем механизм защиты, введем цели переменные, принимающие значения 0, …, *N*.

Связь будет следующей:

$$x_i = s-1,\ s=1,\ldots,N \Leftrightarrow M_{is}=1,$$
$$x_i = N \Leftrightarrow M_{is}=0 \quad \forall s=1,\ldots,N.$$

Задача (8.14) эквивалентна следующей:

$$\max\left\{\sum_{i=1}^{L} F_i(x_i) \,\middle|\, \sum_{i=1}^{L} G_i(x_i) \le C_{\max},\ x_i \in \{0,N\}\right\},$$

где

$$\begin{cases} F_i(x_i) = \begin{cases} \sum\limits_{k=1}^{x_i} \ln\left(1-E_{ik}\right),\ x_i = 1,\ldots,N, \\ 0,\ x_i = 0 \end{cases} \\ G_i(x_i) = \begin{cases} C_{ix_i},\ x_i = 0,\ldots,N-1, \\ 0,\ x_i = N \end{cases} \end{cases}.$$

Общий случай с неидеальными средствами защиты

Рассмотрим общий случай, когда средства защиты не являются абсолютно надежными и каждое средство может использоваться на всех уровнях стека, а не только на одном, как было раньше. В гл. 4 задача (8.13) решалась градиентными методами. После записи задачи (8.13) в виде задачи целочисленного сепарабельного программирования для ее решения могут использоваться уже упомянутый точный метод ПАВ и методы локальной оптимизации (такие, как метод вектора спада и локально-стохастические алгоритмы).

Поскольку задачи дискретного сепарабельного программирования предполагают, чтобы множества значений, которые принимают переменные x_i, отличались для разных i, то перейдем к случаю, когда механизмы защиты могут реализовываться на нескольких уровнях стека, но не обязательно на всех N. Предлагается решать такую задачу с помощью перехода к задаче целочисленного сепарабельного программирования. Для этого необходимо сделать дополнительные вычисления, которые заключаются в следующем. Пусть для каждой i-й киберугрозы существует набор номеров уровней стека $\{s_{i0},\ldots,s_{iv_i}\}$, $v_i < N$, на которых можно реализовать соответствующий механизм защиты.

Тогда целочисленная переменная x_i новой задачи будет принимать значения $0,\ldots,2^{v_i}-1$. В результате этого новую задачу можно записать в виде

$$\min\left\{\sum_{i=1}^{L} F_i(x_i)\,\middle|\; \sum_{i=1}^{L} G_i(x_i) \le C_{\max},\ x_i \in \{0, 2^{v_i} - 1\}\right\}, \qquad (8.15)$$

где значения функций $F_i(x_i)$ и $G_i(x_i)$ легко получить, подставив в i-й множитель целевой функции и i-е слагаемое ограничения задачи (8.13) комбинацию булевых переменных, которой соответствует значение переменной x_i. Полученную задачу можно решать с помощью метода ПАВ и приближенных методов локальной оптимизации (метода вектора спада и локально-стохастических методов). В общем случае, когда средства защиты не являются абсолютно надежными и могут реализовываться на всех уровнях стека, задача (8.15) может иметь достаточно большую размерность, потому что каждая переменная x_i тогда может приобретать $2^N - 1$ значений. На практике многие из этих значений невозможно реализовать, потому размерность множества X_i будет гораздо меньше – не более $2^{N-s} - 1$, где $s < N$ является количеством запрещенных уровней, на которых данное средство защиты нельзя реализовать. Для случая, когда средство может реализовываться не больше, чем на одном уровне, имеем $|X_i| = N + 1$.

Исходную задачу можно решить с помощью точного метода Балаша с модификациями Джеофриона [160], который предполагает, чтобы целевая функция задачи булевого программирования была нелинейной. Единственное, что требуется, – чтобы целевая функция имела вид, который дает возможность однозначно найти лучшее дополнение частичного решения.

Функция для нашего случая, очевидно, является именно такой – для любой булевой переменной значение 1 является лучшим, чем значение 0, если мы максимизируем эту функцию. Пусть имеется частичное решение $\widetilde{M}$, тогда для построения лучшего дополнения $\widetilde{M}$ необходимо всем переменным, не входящим в $\widetilde{M}$, придать значение 1. Нарушение ограничения на этом этапе не учитывается.

Вместо булевой матрицы с компонентами M_{ij}, $i = 1, \ldots, L$, $j = 1, \ldots, N$ будем рассматривать вектор с компонентами $x_{(i-1)N+j} = M_{ij}$. Обозначим как $\overline{Z}$ нижнюю границу значения целевой функции, известную в настоящее время (если в начале работы никакой информации нет, то $\overline{Z} = 0$). Будем хранить координаты частичного решения, которое анализируется на текущем шаге алгоритма, в списке S, точнее, будем заносить в этот список число i, если

$x_i = 1$, и $-i$, если $x_i = 0$. Переменные с индексами, которые не включены в список S (не важно с каким именно знаком), считаются свободными. Также в списке S каждый индекс может быть подчеркнутым, что указывает на то, что для этого частичного решения изменение знака индекса приведет к уменьшению значения целевой функции на всех возможных дополнениях или к нарушению неравенства. Для каждого частичного решения, которое соответствует текущему заполнению списка S, рассматривается его лучшее дополнение x^S, координаты которого для фиксированных переменных устанавливаются в соответствии с индексами из списка, а для свободных – так, чтобы максимизировать целевую функцию, несмотря на возможное нарушение неравенства. Для нашей задачи для максимизации целевой функции все свободные переменные нужно установить в единицу.

Метод Балаша, адаптированный для нашей задачи, будет иметь следующий вид:

1. Пусть $\overline{Z}$ – нижняя граница значения целевой функции. Очистить список *S*.

2. Все переменные x_i, которые не являются свободными, установить согласно заполнению *S*. Все свободные переменные x_i положить равными единице. Сохранить это решение x^S. Вычислить значение целевой функции на этом решении $f(x^S)$.

3. Вычислить величину невязки $y^S = cx^S - C_0$. Если $y^S \le 0$, то перейти к шагу 7.

4. Добавить в список T^S те свободные индексы j, которые удовлетворяют условию $f(x^S \setminus x_j) > \overline{Z}$, где вектор $x^S \setminus x_j$ имеет те же координаты, что и x^S, за исключением одной координаты $x_j = 0$.

5. Вычислить величину $W = y^S - \sum_{j \in T^S} c_j$. Проверяем, выполняется ли $W > 0$. Если это так, то текущее частичное решение нельзя дополнить таким образом, чтобы не нарушалось неравенство, и переходим к шагу 8, где будет сделан переход к другому частичному решению или произойдет остановка алгоритма.

6. Добавить в список S индекс $-j_0$, где $j_0 \in T^S$ и минимизировать величину $W_{j_0} = y^S - c_{j_0}$. Вернуться к шагу 2.

7. Если $f(x^S) > \overline{Z}$, то положить $\overline{Z} = f(x^S)$, а решение x^S сохранить как текущее лучшее.

8. Найти в списке *S* последний неподчеркнутый индекс. Поскольку мы пришли к этому шагу, то улучшить решение нельзя, поэтому надо изменить знак индекса на противоположный и подчеркнуть, чтобы указать, что это решение уже прозондировано. Все индексы после подчеркнутого надо удалить, потому что по ним никаких выводов сделать нельзя. Если такой индекс не найден, то остановиться и выдать последнее лучшее решение с соответствующим значением $\overline{Z}$, иначе вернуться к шагу 2.

Для решения задачи с реализацией средства защиты на одном уровне с помощью метода Балаша также возможно указать, как находить лучшее дополнение частичного решения. Пусть имеется частичное решение $\widetilde{M}$. Известно, что оно не нарушает ни одно неравенство, в частности ограничения на структуру решения. Для того чтобы найти лучшее дополнение $\widetilde{M}$ для всех $i = 1, \ldots, L$, необходимо сделать следующее. Если все переменные $M_{ij} = 0$, $M_{ij} \in \widetilde{M}$, $j = 1, \ldots, N$, причем остается хотя бы одна свободная переменная M_{iq}, то полагаем $M_{iq} = 1$, где q является минимальной среди всех свободных переменных для этого i. Если же среди переменных $M_{ij} \in \widetilde{M}$, $j = 1, \ldots, N$ есть единица, то, исходя из необходимости выполнения ограничения на структуру решения, мы должны положить $M_{iq} = 0$, $q = 1, \ldots, N$ для всех свободных переменных для этого i. Полученное решение и будет лучшим дополнением $\widetilde{M}$ (среди ограничений учтены только ограничения на структуру решения, а стоимость системы на этом этапе не рассматривается).

Другие частные случаи задачи

Рассмотрим различные частные случаи общей задачи (8.13), которые использовались при исследовании градиентного метода в гл. 4 и могут решаться более эффективно с использованием эквивалентных переходов с помощью известных алгоритмов математического программирования.

Рассмотрим случай, когда для каждой i-й киберугрозы существует единственный уровень стека s_i, на котором можно реализовать механизм защиты, а для всех остальных уровней стоимость не определялась (можно

считать ее бесконечной). Показатель эффективности реализации i-й киберугрозы полагаем ненулевым только для s_i-го уровня (того же, на котором возможно реализовать механизм защиты). Полученную нелинейную задачу (8.13) с LN булевыми переменными можно решать с помощью различных методов, например градиентного метода, модифицированного метода Балаша, подобно вышеизложенному, и с помощью методов сепарабельного программирования. Для того чтобы отобразить запрещенность некоторых комбинаций переменных, в методе Балаша достаточно строить частичные решения, начиная с такого, в котором соответствующие переменные будут зафиксированы (в терминах алгоритма это означает включение в список S подчеркнутых индексов). В других методах также принципиально можно учесть ограничения на возможность реализации средств защиты, но проще всего в упомянутом случае привести задачу к эквивалентной задаче ЛБП.

Итак, если в терминах задачи (8.13) имеем

$$E_{ij} = \begin{cases} e_i, & j = s_i \\ 0, & j \neq s_i \end{cases}, \quad C_{ij} = \begin{cases} c_i, & j = s_i \\ \infty, & j \neq s_i \end{cases},$$
$$\alpha_{ij} = 1, \quad i = \overline{1, L}, \quad j = \overline{1, N},$$

то для каждой i-й киберугрозы существует две возможности: использовать механизм защиты или отказаться от противодействия этой угрозе. В первом случае вероятность того, что киберугроза не сможет реализоваться, равна $1-(1-\alpha_{is_i})e_i$, а в другом – $1-e_i$. После логарифмирования зависимость вероятности противодействия i-й киберугрозе будет следующей:

$$\log P_i = (1-x_i)\ln(1-e_i) + x_i \ln\left(1-\left(1-\alpha_{is_i}\right)e_i\right) =$$
$$= \ln(1-e_i) + x_i \ln\left(\left(1-\left(1-\alpha_{is_i}\right)e_i\right)\Big/(1-e_i)\right),$$

где булевая переменная x_i обозначает начальную переменную M_{is_i}. Для определения конфигурации оптимальной КС КЗИ достаточно решить следующую задачу ЛБП:

$$\begin{cases} \sum_{i=1}^{L} x_i \ln\left(\left(1-\left(1-\alpha_{is_i}\right)e_i\right)/(1-e_i)\right) \to \max\limits_{x_i \in \{0,1\}} \\ \sum_{i=1}^{L} c_i x_i \le C_0 \end{cases}. \qquad (8.16)$$

Подобной также будет ситуация, когда механизмы можно реализовать на всех уровнях, но каждая i-я киберугроза имеет ненулевую эффективность

реализации лишь на одном уровне стека протоколов (обозначим его номер s_i). В этом случае опять достаточно решить задачу ЛБП, приведенную выше.

Результаты численных экспериментов

Для проверки работоспособности разработанных алгоритмов и доказательства корректности эквивалентных задач был проведен ряд численных экспериментов. В пяти случаях это были те же модельные примеры, которые использовались для исследования алгоритмов, описанных в гл. 4, и еще один был сгенерирован дополнительно. Для всех примеров средства защиты считались абсолютно надежными.

В качестве первого примера рассматривалась киберсистема, эффективность реализации всех киберугроз в которой была одинаковой и равнялась 0,05, причем ненулевой она была только для одного уровня стека протоколов для каждой киберугрозы. Что касается матрицы ценностей, то ее элементы для *i*-го средства защиты определялись только для одного уровня стека – того, на котором можно реализовать это средство. Количество киберугроз и механизмов защиты составляло $L = 20$, а количество уровней стека $N = 7$. Максимальная стоимость средств защиты, входящих в состав КС КЗИ, не могла превышать величины $C_{\max}$, составляющей 50 % от суммарной стоимости всех средств защиты. Для нахождения оптимальной конфигурации КС КЗИ в этом случае достаточно было решить задачу ЛБП вида (8.16). Для ее решения использовались алгоритм Балаша и метод вектора спада, реализованные в виде процедур библиотеки численного анализа НИВЦ МГУ [162] (процедуры MNR3R и MLC6R).

По результатам экспериментов можно сказать следующее. Точный метод Балаша нашел тот же глобальный максимум, что и градиентный метод, в то время как метод вектора спада нашел только локальный. Относительная погрешность полученного локального максимума составила 5 % (0,6983 вместо точного значения 0,7351). Начальная вероятность сохранения безопасности при полном отказе от реализации средств защиты составляла $(1 - 0{,}05)^L = 0{,}3585$.

В качестве второго и третьего примеров рассматривалась киберсистема, эффективности реализации киберугроз в которой отличались, причем снова ненулевыми они были только для одного уровня стека протоколов для каждой киберугрозы. Данные примеры отличались величиной максимальной стоимости

КС КЗИ, соответственно 50 и 93,4 % от суммарной стоимости всех средств защиты. Применение упомянутых алгоритмов решения задачи ЛБП позволило в обоих случаях найти тот самый максимум (локальный совпал с глобальным), который был найден в гл. 4 с помощью градиентного метода. По сравнению с начальной вероятностью сохранения безопасности при полном отказе от реализации средств защиты $P_0 = \Pi(1 - e_i) = 0{,}3547$, найденные конфигурации КС КЗИ позволили повысить вероятность в соответствии со значениями $P_1 = 0{,}9136$ и $P_2 = 0{,}9978$.

В качестве четвертого примера рассматривалась киберсистема, в которой каждая киберугроза имела ненулевые вероятности реализации на трех уровнях стека протоколов из имеющихся $N = 7$, а на всех других уровнях средства защиты не могли реализовываться вообще. Количество киберугроз и средств защиты снова составляло $L = 20$. Стоимость реализации средств защиты была обратно пропорциональной номеру уровня стека. Ограничение стоимости устанавливалось на уровень 19 % от стоимости конфигурации, в которой для каждой киберугрозы механизм защиты реализовывался на самом низком возможном уровне. Оптимальная конфигурация КС КЗИ вычислялась с помощью решения задачи сепарабельного программирования с помощью локально-стохастического алгоритма [161]. По сравнению с решением, найденным в гл. 4, после повторного запуска алгоритма с разных начальных точек была найдена конфигурация КС КЗИ, которая обеспечивает примерно эквивалентный уровень защиты – вероятность составляла $P' = 0{,}9516$, в отличие от значения $P = 0{,}9598$, найденного с помощью градиентного метода.

В качестве пятого примера рассматривалась киберсистема, в которой каждая киберугроза имела ненулевые вероятности реализации на трех уровнях стека протоколов из имеющихся $N = 7$, но средства защиты могли реализовываться на любом уровне. С помощью неоднократного запуска локально-стохастического алгоритма с разных начальных точек удалось улучшить решение, найденное с помощью градиентного метода. Полученное значение вероятности сохранения безопасности $P' = 0{,}947$ оказалось лучше найденного ранее $P = 0{,}901$.

Вероятностные характеристики киберугроз для шестого примера были сгенерированы случайным образом, стоимость средств была обратно пропорциональной номеру уровня стека. Стек протоколов содержал $N = 4$ уровня, а количество киберугроз и средств защиты составляло $L = 20$.

Характер изменения вероятности сохранения защищенности при решении задач 4–6 в процессе работы локально-стохастического алгоритма можно проследить на рис. 8.7.

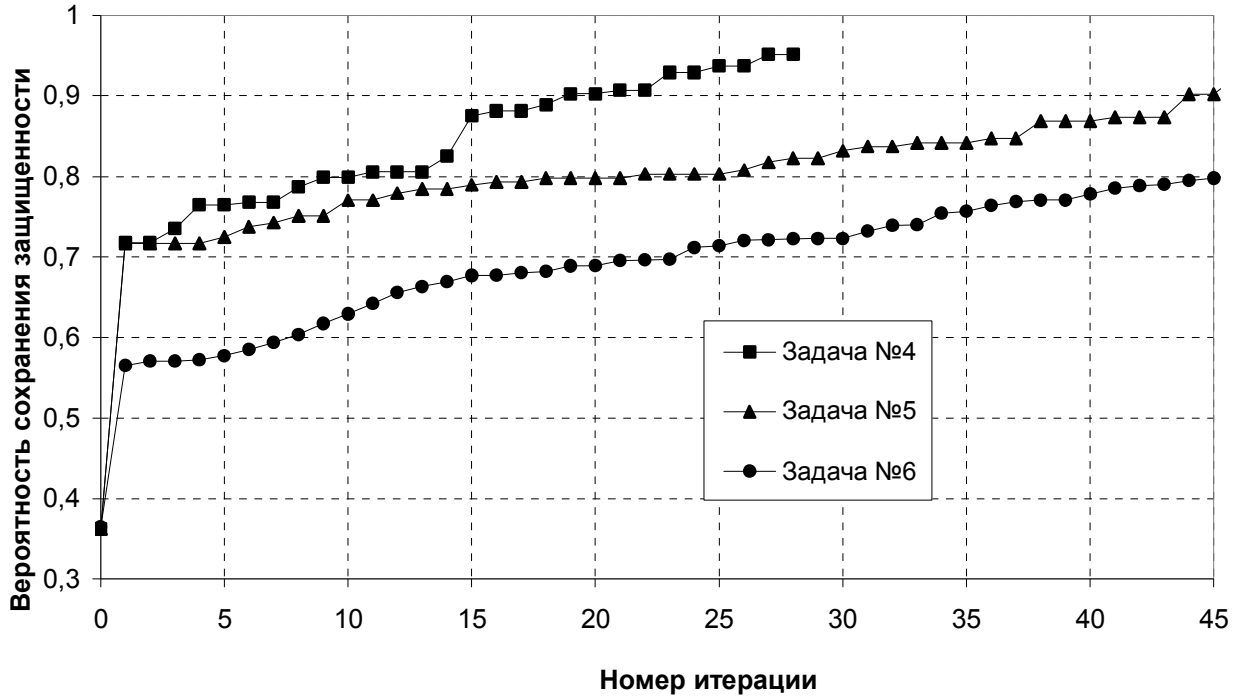


Рис. 8.7. Сходимость алгоритмов при решении задач 4–6

Выводы

Изложенные математическая модель, задача математического программирования и алгоритм ее решения могут быть использованы для синтеза КС КЗИ с минимальной стоимостью механизмов защиты, которые принадлежат к классу открытых и при этом обеспечивают гарантированный остаточный риск в ИКС. С использованием логико-вероятностных методов теории надежности структурно сложных систем удалось получить численную зависимость потерь от кибератаки. Эта зависимость была использована для постановки соответствующей задачи математического программирования, которая является формальным выражением требований, выдвигаемых к КС КЗИ.

Полученная задача принадлежала к классу задач нелинейного булевого программирования, и из-за особенностей, которые следовали из практического опыта и наблюдений, были предложены два способа ее решения: первый – переход к эквивалентной линейной задаче в виде задачи сепарабельного программирования, что позволило уменьшить размерность задачи; второй –

задача линейного булевого программирования решалась прямо, без перехода к эквивалентной задаче сепарабельного программирования. Имея одинаковую асимптотическую сложность, оба алгоритма могут считаться альтернативными, и на практике можно проводить переход от использования одного из них к другому при превышении определенной предельной размерности задачи.

Также рассмотрена задача синтеза КС КЗИ с минимальной стоимостью с учетом эффективности механизмов защиты. Разработан алгоритм решения задачи целочисленного сепарабельного программирования с использованием локально-стохастических алгоритмов. При сравнении его с точным *P*-методом и методом вектора спада для решения задачи ЛБП он показал лучшие показатели и может быть использован как составляющая системы автоматизированного проектирования КС КЗИ, оптимальной по экономическому критерию.

Все разработанные в данной главе алгоритмы подтвердили свою работоспособность и эффективность и могут быть использованы при проектировании реальных систем КС КЗИ, удовлетворяющих таким требованиям, как использование технологии систем с открытой архитектурой, и для которых реально выполнить численную оценку вероятностных характеристик киберугроз безопасности и потерь от реализации этих киберугроз.

9. ЛОГИКО-ВЕРОЯТНОСТНЫЙ ПОДХОД К ПОСТРОЕНИЮ СИСТЕМ КИБЕРНЕТИЧЕСКОЙ ЗАЩИТЫ

В данной главе рассматривается процедура построения защищенной ИКС, базирующаяся на этапах создания автоматизированных систем, этапах построения корпоративной сети и порядке создания комплексной системы защиты информации. На основе данной процедуры предлагается логико-вероятностный подход к построению защищенной ИКС, который включает в себя предложенные в предыдущих главах модели, методы и алгоритмы.

9.1. Этапы построения защищенной ИКС

Как уже упоминалось, для защиты ИКС от разного вида киберугроз создается комплексная система кибернетической защиты информации – КС КЗИ.

Для обеспечения методологического подхода к построению системы кибернетической защиты информации в компьютерных системах могут использоваться международные (ISO) [74, 75] и отечественные стандарты (ДСТУ) [76, 77, 78], а также нормативные документы (НД ТЗИ) [9, 23, 79, 80].

В своем большинстве подобные методологические подходы представляют собой структурированное описание последовательных задач и документов, которые необходимо подготовить для построения системы защиты, и нацелены на дальнейшую процедуру международной или государственной сертификации, или аттестации.

Но, как уже говорилось, даже четкое следование этим стандартам и нормативным документам не будет гарантировать отсутствие проблем во время построения системы кибернетической защиты информации.

В соответствии с [81], порядок создания комплексной системы кибернетической защиты информации в ИКС является независимым от того, находится ли данная система на стадии проектирования, эксплуатируется либо модернизируется. Но, исходя из практического опыта авторов и концепции глубокой эшелонированной защиты (defense-in-depth) [82], необходимо, чтобы комплексная система кибернетической защиты информации начинала создаваться на стадии формирования требований к ИКС, во время ее проектирования. Откладывание создания системы защиты на более поздние этапы может

привести к дополнительным расходам и менее эффективным решениям, и, возможно, в худшем случае, к невозможности достичь необходимого уровня защищенности (безопасности) [83].

Исходя из этого, начинать проектировать КС КЗИ (или хотя бы задумываться о ней) есть смысл как можно раньше, а также обязательно интегрировать ее разработку в общий процесс проектирования ИКС, как это предлагается в методологии Security Development Lifecicle (SDL) [58].

Но для начала проектирования КС КЗИ необходим набор входных данных и условий, которые будут появляться как результаты соответствующих этапов проектирования самой ИКС. Кроме того, ряд решений, принятых при проектировании КС КЗИ, также будут влиять на проектируемую архитектуру ИКС, приводя к необходимости пересмотра ее структуры, архитектуры приложений, сегментации сети и топологии. Для этого детальнее остановимся и рассмотрим взаимосвязь между этапами проектирования ИКС и КС КЗИ.

Как уже отмечалось в п. 1.1, ИКС состоит из информационной системы, средств коммуникации и управления информационными потоками, а также сети и каналов передачи информации. В зависимости от размера и требований к функциональности информационно-коммуникационной системы, для каждой из ее составляющих может существовать отдельная последовательность этапов их создания, но глобально эти этапы будут соответствовать стадиям создания АС. Рассмотрим их детальнее, больше внимания уделяя этапам проектирования [163]:

1. Формирование требований к АС:

1.1. Обследование объекта и обоснование необходимости создания АС.

1.2. Формирование требований пользователей к АС.

1.3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания).

2. Разработка концепции АС:

2.1. Изучение объекта.

2.2. Проведение необходимых научно-исследовательских работ.

2.3. Разработка вариантов концепции АС, удовлетворяющих требованиям пользователей.

2.4. Оформление отчета о выполненной работе.

3. Техническое задание.

4. Эскизный проект:

4.1. Разработка предварительных проектных решений по системе и ее частям.

4.2. Разработка документации на АС и ее части.

5. Технический проект:

5.1. Разработка проектных решений по системе и ее частям.

5.2. Разработка документации на АС и ее части.

5.3. Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку.

5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации.

6. Рабочая документация.

7. Ввод в действие.

8. Сопровождение АС.

Основываясь на общих этапах создания АС, покажем взаимосвязь между этапами проектирования КС КЗИ и этапами проектирования корпоративной сети.

В качестве основы для описания этапов проектирования защищенной ИКС возьмем этапы создания корпоративной сети. В данные этапы входят все основные составные части ИКС: информационная система, средства коммуникации и управления информационными потоками, сети и каналы передачи данных.

Рассмотрим детально этапы построения корпоративной сети. Согласно [164, 165] построение корпоративной сети состоит из следующих этапов:

1. Анализ требований к корпоративной сети:

1.1. Цели создания корпоративной сети.

1.2. Требования к функциям корпоративной сети.

1.3. Анализ существующих систем.

1.4. Критерии качества для оценки эффективности сети.

2. Разработка бизнес-модели организации (предприятия):

2.1. Основные, административные и вспомогательные бизнес-процессы организации.

2.2. Подразделения, структура и иерархические взаимоотношения между подразделениями.

2.3. Информационные потоки в организации и между подразделениями.

2.4. Группы пользователей системы (сети).

2.5. Основные сервисы корпоративной сети.

2.6. Архитектура приложений и информационных систем.

3. Разработка технической модели корпоративной сети (структурный синтез):

3.1. Логическая структуризация корпоративной сети:

3.1.1. Выделение сегментов и отдельных групп.

3.1.2. Установление связей и информационных потоков между группами.

3.2. Требования к аппаратному и программному обеспечению.

4. Разработка физической модели корпоративной сети (параметрический синтез).

5. Разработка технического задания.

6. Моделирование и оптимизация сети.

7. Установка и налаживание системы.

8. Тестирование системы.

9. Сопровождение и эксплуатация системы.

Теперь приведем этапы создания комплексной системы защиты и рассмотрим их связь с этапами построения корпоративной сети.

Для обеспечения защиты информации в ИКС создается комплексная система защиты информации. Порядок создания комплексной системы защиты информации, согласно НД ТЗИ [81], является следующим:

1. Формирование требований к КСЗИ:

1.1. Обоснование необходимости создания КСЗИ.

1.2. Обследование среды функционирования ИКС.

1.3. Перечень объектов защиты.

1.4. Разработка модели угроз.

1.5. Формирование задачи на создание КСЗИ.

2. Разработка политики безопасности:

2.1. Разработка концепции безопасности информации в АС.

2.2. Анализ рисков.

2.3. Определение требований к мероприятиям, методам и средствам защиты.

2.4. Выбор основных решений для обеспечения безопасности информации.

2.5. Организация выполнения восстановительных работ и обеспечение непрерывного функционирования АС.

2.6. Документальное оформление политики безопасности.

3. Разработка технического задания (ТЗ) на создание КСЗИ:

3.1. Описание и классификация ресурсов системы, для которой строится КСЗИ:

3.1.1. Операционные системы, которые используются.

3.1.2. Средства коммуникации.

3.1.3. Категории информации, ее вид, форма хранения, технология обработки.

3.1.4. Персонал и пользователи.

3.1.5. Территория, на которой расположена система.

3.2. Разработка информационной модели для системы:

3.2.1. Формальное или неформальное описание информационных потоков.

3.2.2. Описание пользовательских интерфейсов.

3.3. Определение перечня угроз и потенциальных каналов утечки информации.

3.4. Оценка рисков.

3.5. Определение услуг безопасности, которые нужно реализовать.

3.6. Обоснование необходимости проведения спецпроверок и специсследований средств вычислительной техники и других технических средств, а также специального оборудования помещений.

3.7. Требования к мероприятиям защиты, которые дополняют программно-технические средства (организационные, физические).

3.8. Требования относительно оборудования, приборов и метрологического обеспечения работ.

3.9. Перечень макетов, стендов (если такие разрабатываются).

3.10. Оценка стоимости, эффективности избранных средств.

3.11. Принятие окончательного решения о составе КСЗИ.

4. Разработка проекта КСЗИ.

5. Введение КСЗИ в действие.

6. Сопровождение КСЗИ.

Во время анализа требований к корпоративной сети для эффективной защиты ИКС должно быть принято решение относительно необходимости разработки КС КЗИ и определены основные требования к ней.

После того, как разработана бизнес-модель организации, становятся известными основные сервисы корпоративной сети, архитектура приложений и информационных систем, информационные потоки и основные группы пользователей. Эти данные будут входными для этапа обследования среды функционирования ИКС при создании КС КЗИ.

Также на основе этих данных и требований к функциям корпоративной сети может быть определен перечень объектов защиты, разработана модель киберугроз и сформирована задача на создание КС КЗИ.

Дальше для построения защищенной ИКС необходимо разработать политику безопасности и уже на ее основе выполнять разработку технической модели корпоративной сети, а именно: выполнять проектирование логической структуры сети и разрабатывать требования к необходимому аппаратному и программному обеспечению (включая активное сетевое оборудование, межсетевые экраны, специальное программное обеспечение). При этом может быть необходимость возвратиться к предыдущим этапам разработки бизнес-модели организации и политики безопасности для согласования их требований между собой.

После нескольких таких итераций должна быть зафиксирована и документально оформлена политика безопасности, а также описаны ресурсы и разработана информационная модель для системы.

Основываясь на этом, можно переходить к разработке технического задания на создание КСЗИ и разработке технического задания на разработку корпоративной сети.

Покажем, как разработанные в предыдущих главах модели и алгоритмы могут быть использованы во время построения КС КЗИ.

9.2. Логико-вероятностный подход к построению защищенных ИКС

В гл. 2–8 на основе логико-вероятностного метода были предложены следующие модели и алгоритмы:

1. Структурная логико-вероятностная модель оценки комплексного уровня защищенности ИКС:

1.1. Вероятностная функция оценки успешности кибератаки.

1.2. Структурная значимость объектов/компонентов ИКС.

1.3. Построение безопасной топологии сети ИКС.

1.4. Оптимальное размещение механизмов защиты в узлах ИКС.

2. Иерархическая логико-вероятностная модель защищенной ИКС:

2.1. Вероятностная функция оценки уровня кибернетической безопасности ИКС.

2.2. Оптимальное размещение механизмов защиты по уровням стека протоколов взаимодействия компонентов ИКС:

2.2.1. Синтез кибернетической системы защиты информации с минимальной стоимостью механизмов защиты с учетом эффективности механизмов защиты.

2.2.2. Синтез кибернетической системы защиты информации с максимальной вероятностью сохранения безопасности.

На их основе можно рекомендовать общую процедуру построения КС КЗИ, состоящую из следующих шагов:

1. Определение уровня защищенности ИКС и ее объектов на основе функции вероятности успешности кибератаки.

2. Определение структурной значимости объектов ИКС.

3. Построение безопасной топологии сети ИКС.

4. Оптимальное размещение механизмов защиты в узлах ИКС.

5. Определение кибернетической безопасности ИКС на основе вероятностной функции оценки уровня кибернетической безопасности.

6. Оптимальное размещение механизмов защиты по уровням стека протоколов взаимодействия компонентов ИКС в соответствии с требуемыми показателями (максимальной вероятностью сохранения безопасности, минимальной стоимостью механизмов защиты).

Основываясь на порядке создания КСЗИ и этапах построения корпоративной сети, рассмотрим в совокупности применение данных шагов для построения КС КЗИ. Для этого определим входные данные для каждой из приведенных процедур и этапы разработки, на которых они могут быть применены.

Как было указано, первый (*анализ требований к корпоративной сети*) и второй (*разработка бизнес-модели организации*) этапы построения *корпоративной сети* будут предоставлять необходимую информацию для того, чтобы начать *обследование среды* функционирования ИКС. Также на этих этапах становится известным перечень объектов ИКС $V = \{v_1,...,v_N\}$, а также зависимость объектов (сервисов) один от другого, территориальное размещение объектов и организационная структура. На основе этого проектируется начальный вариант топологии сети G.

После *обследования среды* функционирования может быть выделен перечень объектов киберзащиты $O = (o_1,...,o_M) \subset V$, а на основе модели киберугроз и модели нарушителя – источники кибератак $A = (a_1,...,a_K) \subset V$.

При определении множества O неявно проводится анализ рисков и выбираются те объекты, для которых ущерб от кибератак будет наибольшим, при этом предполагается приблизительное равенство указанного ущерба в случае успешной кибератаки.

Исходя из этих данных, которые получены на этапе *обследования среды* функционирования, строятся граф кибератак $G(V,E,A,O)$ и функция вероятности успешности кибератаки (5.17), на основе которой можно определить структурную значимость объектов ИКС $\xi = (\xi_1,...,\xi_N)$ – (5.34).

Полученные значения структурной значимости объектов $\xi = (\xi_1,...,\xi_N)$ используем во время этапа *разработки политики безопасности*, а именно когда выполняется *анализ рисков*. По значению структурной значимости можно определить перечень объектов, кибератака на которые при данной топологии сети будет иметь наибольшее влияние на кибербезопасность всей ИКС. Именно для этих объектов нужно будет впоследствии выполнять детальный мониторинг и прорабатывать стратегию защиты.

Также во время *анализа рисков* должны быть определены значения вероятностей захвата объектов $V = \{v_1,...,v_N\}$ ИКС $(P_1,...,P_N)$ ((5.7), (5.8), (5.9)).

Эти вероятности захвата будут использоваться во время дальнейших расчетов функции вероятности успешности кибератак.

На основе вероятностей захвата объектов с помощью функции вероятности успешности кибератак может быть вычислена вероятность успешности кибератаки на киберсистему. Полученные значения вероятностей могут быть использованы для оценки рисков.

Пользуясь вероятностями захвата объектов $(P_1,...,P_N)$, вычисляется начальное значение функции вероятности успешности кибератак:

$$P_{\max} = P(G, A, O, P_1,..., P_N)$$

Это значение будет максимальным значением вероятности успешности кибератаки на ИКС. В результате построения системы защиты (оптимизации топологии и применения механизмов защиты) это значение должно уменьшиться до некоего P_{rez} – такого значения функции вероятности, что $P_{rez} \le P_{\max}$.

После *анализа рисков* осуществляется *определение требований* к мероприятиям, методам и средствам защиты. На этом этапе, в зависимости от уровня конфиденциальности информации, которая обрабатывается в ИКС, и ее критичности (а также в зависимости от ущерба, причиненного реализацией кибератак, финансового состояния владельца), и других факторов, определяются приемлемые затраты для достижения необходимого уровня защищенности, т. е. определяется максимальная стоимость системы защиты $C_{\max}$ или выставляется требуемый уровень защищенности (приемлемый риск) – $R_{\max}$.

Во время *выбора основных решений* для обеспечения безопасности информации в качестве одного из средств киберзащиты на основе политики безопасности и других факторов выполняется выделение сегментов в сети ИКС.

Кроме политики безопасности, при выделении сегментов должны также учитываться принципы, приведенные в гл. 6. Таким образом, этап *выбора основных решений* для обеспечения безопасности информации будет связан с этапом *логической структуризации* корпоративной сети. В результате, на основе этих двух этапов, получим множество сегментов сети $S = \{S_1, S_2,..., S_T\}$, в которых должны размещаться объекты системы $V = \{v_1,..., v_N\}$.

На основе этих данных может быть осуществлено распределение объектов системы по сегментам сети, т. е. построение безопасной топологии сети.

Для этого должно быть построено множество приемлемых вариантов топологии $O_S(G)$ и среди них выбрана лучшая с точки зрения информационной безопасности:

$$\begin{cases} G^* = \arg\min\limits_{G \in O_S} P(G) \\ O_S(G) = \{G_1, G_2, ..., G_k\} \subseteq \Omega \end{cases}.$$

По результатам вместо первоначальной топологии сети G получим топологию G^*, где $P(G^*) \le P(G) = P_{\max}$. На основе топологии G^* в дальнейшем будет строиться функция вероятности успешности кибератаки, а также будет осуществлено документальное оформление политики безопасности.

На основе новой функции вероятности успешности кибератаки, которая построена на основе топологии сети G^*, может быть записана функция чувствительности для объектов системы и подсчитана значимость объектов $\xi^* = (\xi^*_1, ..., \xi^*_N)$ (5.28) с использованием значений вероятностей захвата объектов $(P_1, ..., P_N)$.

Полученные здесь значимости объектов могут служить для определения критических с точки зрения безопасности объектов, перечень которых составляется на этапе *организации выполнения* восстановительных работ и обеспечения непрерывного функционирования АС.

Заключительным во время этапа *разработки политики безопасности* является *документальное оформление* политики безопасности.

Во время *разработки технического задания* должны быть определены средства защиты и их эффективность. Для этого необходимо решить задачу оптимального размещения механизмов защиты в узлах ИКС. Для этого для каждого объекта системы v_i еще раз могут быть оценены вероятности его захвата без применения механизмов защиты $P = \{P_1, P_2, ..., P_N\}$.

Дальше необходимо задать механизмы защиты M_i, их коэффициенты эффективности K_i и стоимость C_{M_i}, которые могут быть применены для уменьшения вероятности (вероятностей) захвата объектов P_i. Основываясь на этих данных для фиксированной топологии сети G^* и учитывая максимальную стоимость системы защиты информации $C_{\max}$, может быть решена задача оптимального размещения механизмов защиты в узлах ИКС

$$\begin{cases} M^* = \arg\min\limits_{M} P_{str}(E, M, K) \\ C(M) = \sum\limits_{i} C_{M_i} \le C_{mzx} \end{cases}$$

и оптимального размещения механизмов защиты по уровням стека протоколов взаимодействия компонентов ИКС (п. 8.8)

$$\begin{cases} M^* = \arg\min\limits_{M} R_{hie}(M) \\ C(M) = \sum\limits_{i} C_{M_i} \le C_{mzx} \end{cases}$$

или (п. 8.7)

$$\begin{cases} R_{hie}(M) \le R_{\max} \\ M* = \arg\min\limits_{M} C(M) \cdot \end{cases}$$

Результатом решения задачи будут множество механизмов защиты M^* и уровень защищенности $P(G^*, M^*)$ (значение функции вероятности успешности кибератаки), которого можно достичь при данной стоимости системы защиты информации.

В зависимости от затрат на систему защиты информации $C_{\max}$ значение функции вероятности успешности кибератаки будет находиться в промежутке между $P_{\min}$ и $P(G^*)$:

$$P_{\min} \le P(G^*, M^*) \le P(G^*),$$

где $P_{\min}$ – минимальное значение функции вероятности успешности кибератаки, которое будет достигаться без учета ограничения на стоимость системы защиты и с применением всех механизмов защиты – $P_{\min} = P(G^*, M)\big|_{M=(1,1,\ldots,1)}$;

$P(G^*)$ – значение функции вероятности успешности кибератаки без применения механизмов защиты (или при $C_{\max} = 0$) – $P(G^*) = P(G^*, M)\big|_{M=(0,0,\ldots,0)}$.

Таким образом, по результатам оптимального размещения механизмов защиты, значение функции вероятности успешности кибератаки будет равно $P_{res} = P(G^*, M^*)$. Таким образом, будет иметь место следующее соотношение:

$$P_{\min} \le P_{res} \le P(G^*) \le P_{\max}$$

Разность между начальным значением вероятности успешности кибератаки $P_{\max}$ и полученным P_{res} будет показывать, на сколько увеличилась защищенность ИКС в результате внедрения КС КЗИ.

После разработки проекта КС КЗИ и введения его в действие начинается эксплуатация ИКС, во время которой нужно продолжать отслеживать значение функции вероятности успешности кибератаки. Это может выполняться в рамках сопровождения КС КЗИ или после обнаружения кибератаки. Во время сопровождения КС КЗИ необходимо выполнять периодическую переоценку входных данных параметров P, K, анализировать возможности появления новых источников киберугроз A и объектов кибератак O, а также отслеживать изменения топологии сети G.

Кроме того, со временем значение функции вероятности успешности кибератаки будет начинать увеличиваться из-за того, что в объектах системы будут проявляться новые уязвимости, что приведет к увеличению вероятности захвата объектов P. В свою очередь, механизмы защиты, которые используются, будут становиться менее эффективными из-за появления новых кибератак, что формально приведет к уменьшению их коэффициентов эффективности K. Таким образом, будет справедливым соотношение

$$P(P_1,...,P_n;K_1,...,K_n;t_1) \le P(P_1,...,P_n;K_1,...,K_n;t_2), \text{ при } t_1 \le t_2.$$

Исходя из этого, задачей сопровождения КС КЗИ будет не допускать, со временем, существенного увеличения значения функции вероятности успешности кибератаки $P(P_1,...,P_n;K_1,...,K)$:

$$P(E_1,...,E_n;K_1,...,K;t) - P_{res} \le \Delta.$$

Выводы

В данной главе предложен логико-вероятностный подход к построению защищенных ИКС, который заключается в последовательном применении совокупности приведенных в работе моделей и методов на каждом из этапов построения кибернетической системы. Также предлагается критерий для постоянного отслеживания уровня защищенности во время функционирования системы.

ВЫВОДЫ

Данная монография посвящена моделям и методам кибернетической защиты ИКС, построенным на основе логико-вероятностного метода.

Для получения результатов работы был проведен анализ современных ИКС, рассмотрены основные характеристики, а также факторы, которые влияют на уровень их защищенности. За основу были взяты базовые теоретические и прикладные модели безопасности, теории защиты информации. Проанализированы существующие способы для оценки уровня защищенности ИКС. На основе этого предложены различные варианты постановок задач синтеза систем кибернетической защиты информации, а также сделан вывод о необходимости наличия функции оценки уровня защищенности ИКС от кибератак в явном виде.

Много внимания уделено построению функций для оценки уровня защищенности ИКС и вероятности успешности кибератаки на ИКС, на основе которых решаются различные задачи синтеза систем кибернетической защиты информации.

Главными результатами работы являются:

– разработанные логико-вероятностная математическая модель и функция для оценки уровня защищенности ИКС позволяют оценивать уровень кибернетической безопасности ИКС при воздействии на нее заданного множества кибернетических угроз, отличающаяся учетом условий и выявленных закономерностей функционирования ИКС с открытой архитектурой;

– на основе функции оценки уровня защищенности ИКС разработан набор методов для синтеза систем кибернетической защиты информации с учетом ограничений на стоимость средств защиты, для синтеза систем кибернетической защиты с минимальной стоимостью механизмов защиты, при этом обеспечивающих гарантированный остаточный риск, а также для синтеза систем кибернетической защиты с минимальной стоимостью с учетом эффективности механизмов защиты;

– построены логико-вероятностные математическая модель и функция вероятности успешности кибератаки, которая учитывает топологию размещения объектов ИКС и информационные потоки между ними, а также определена функция чувствительности для определения структурной значимости объектов ИКС;

– на основе функции вероятности успешности кибератаки разработаны метод для построения безопасной топологии сети ИКС и метод для оптимального размещения механизмов защиты при ограничении на стоимость системы киберзащиты;

– предложен логико-вероятностный подход к построению защищенных ИКС, который заключается в последовательном применении совокупности приведенных в работе моделей и методов на каждом из этапов построения кибернетической системы.

Часть решенных задач синтеза систем кибернетической защиты принадлежит к классу задач нелинейного дискретного, булевого и сепарабельного программирования. Для решения этих задач в работе приведены соответствующие алгоритмы.

Были выполнены вычислительные исследования, результаты которых подтвердили конструктивность разработанных математических моделей, правильное отражение в них выявленных условий и закономерностей функционирования ИКС при реализации кибернетических угроз и их предотвращения механизмами защиты.

Все разработанные методы и алгоритмы подтвердили свою работоспособность и эффективность и могут быть использованы при проектировании реальных систем кибернетической защиты.

СПИСОК ЛИТЕРАТУРЫ

1. *Стратегія* забезпечення кібернетичної безпеки України. (Проект) [Електронний ресурс]. – Режим доступу : http://www.niss.gov.ua/public/File/2013_nauk_an_rozrobku/.

2. *Грекул В. И.* Проектирование информационных систем / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – М. : Изд-во «ИНТУИТ.РУ "Интернет-Ун-т Информ. Технологий"», 2008. – 304 с.

3. *Wikipedia*. Інформаційна система [Електронний ресурс]. – Режим доступу : http://uk.wikipedia.org/wiki/Інформаційна_система

4. *Грайворонський М. В.* Безпека інформаційно-комунікаційних систем / М. В. Грайворонський, О. М. Новіков. – Київ : BHV, 2009. – 608 с.

5. *Microsoft*. Руководство Microsoft по проектированию архитектуры приложений [Электронный ресурс]. – 2-е изд. – Режим доступа : http://apparchguide.ms/

6. *Microsoft*. Network Architecture Blueprint. Windows Server System Reference Architecture [Electron. resource]. – Access link : http://www.microsoft.com/wssra

7. *Sun Microsystems*. Software Products [Electron. resource]. – Access link : http://www.sun.com/software/

8. *ISO*/*IEC* 8649-50(3):1986, Information technology – Open Systems Interconnection – VT Service. Protocol specification.

9. *НД ТЗІ* 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу.

10. *FIPS PUB* 146-1, Federal Information Processing Standards Publication. (Supersedes FIPS PUB 146-1988 August 24).U.S. Department of Commerce, National Institute of Standards and Technology (USA). April 1991.

11. *ISO*/*IEC* 7498-2:1989, Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture.

12. *IEEE802.10B*. IEEE Standards for Interoperable Local Area Network (LAN) Security (SILS): Part B – Secure Date Exchange. – April 1992.

13. *ISO*/*IEC* 10181-1:1996, Information technology – Security frameworks for open systems: Overview.

14. *ISO*/*IEC* 10181-2:1996, Information technology – Security frameworks for open systems: Authentication framework.

15. *ISO*/*IEC* 10181-3:1996, Information technology – Security frameworks for open systems: Access control framework.

16. *ISO*/*IEC* 10181-4:1996, Information technology – Security frameworks for open systems: Non repudation framework.

17. *ISO*/*IEC* 10181-5:1996, Information technology – Security frameworks for open systems: Confidentiality framework.

18. *ISO*/*IEC* 10181-6:1996, Information technology – Security frameworks for open systems: Integrity framework.

19. *ISO*/*IEC* 10181-7:1996, Information technology – Security frameworks for open systems: Security audit framework.

20. *Демченко Ю. В.* Безопасность компьютерных сетей в соответствии с рекомендациями стандарта ISO 7498-2 / Ю. В. Демченко // Безопасность информации. – 1995. – № 3. – С. 23–35.

21. *Демченко Ю. В.* Архитектура безопасности Интернет и компьютерных сетей на основе протоколов TCP/IP / Ю. В. Демченко // Безопасность информации. – 1996. – № 3. – С. 34–48.

22. *НД ТЗИ* 1.1-002-99. Общие положения по защите информации в компьютерных системах от несанкционированного доступа.

23. *НД ТЗИ* 2.5-004-99. Критерии оценки защищенности информации в компьютерных системах от несанкционированного доступа.

24. *Можаев А. С.* Общий логико-вероятностный метод анализа надежности сложных систем : учеб. пособие / А. С. Можаев. – Л. : ВМА, 1988. – 68 с.

25. *ISO*/*IEC* 7498-1:1994, Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.

26. *ISO*/*IEC* 8322-3:1985, Information technology – Open Systems Interconnection – Session Service. Protocol specification.

27. *ISO*/*IEC* 8326-7:1985, Information technology – Open Systems Interconnection – Session Service. Protocol specification.

28. *ISO*/*IEC* 9571-1-4:1985, Information technology – Open Systems Interconnection – FTAM Service. Protocol specification.

29. *ISO*/*IEC* 8649-50(2):1986, Information technology – Open Systems Interconnection – ASCE Service. Protocol specification.

30. *Harrison M.* Protection in Operating Systems / M. Harrison, W. Ruzzo, J. Ullman // Communications of the ACM. – 19 (8). – 1976.

31. *Халсалл Ф.* Передача данных, сети компьютеров и взаимосвязь открытых систем / Ф. Халсалл ; пер. с англ. – М. : Радио и связь, 1995. – 408 с.

32. *Боня Ю. Ю.* Синтез систем захисту інформації з мінімальною вартістю механізмів захисту / Ю. Ю. Боня, О. М. Новіков // Проблеми керування та інформатики. – 2006. – № 3. – С.147–156.

33. *Герасименко В. А.* Защита информации в АСОД: в 2 т. / В. А. Герасименко. – М. : Энергоатомиздат, 1994. – 400 с.

34. *Девянин П. Н.* Модели безопасности компьютерных систем : учеб. пособие для студ. вузов / П. Н. Девянин. – М. : Академия, 2005. – 144 с. – ISBN 5-7695-2053-1.

35. *Теория* и практика обеспечения информационной безопасности / под ред. Д. П. Зегжды. – М. : Яхтсмен, 1996. – 297 с.

36. *Зегжда Д. П.* Как построить защищенную информационную систему. / Д. П. Зегжда, А. М. Ивашко. – СПб. : НПО «Мир и семья-95», 1998. – Ч. II. – 256 с.

37. *Хоффман Л. Д.* Современные методы защиты информации / Л. Д. Хоффман. – М. : Советское радио, 1980. – 286 с.

38. *Qian Y.* Information Assurance: Dependability and Security in Networked Systems / Y.Qian, D. Tipper, P. Krishnamurthy, J. Joshi. – Morgan Kaufmann. – 2007. – 576 p.

39. *Manadhata P.K.* An Approach to Measuring A System's Attack Surface / Manadhata P.K., Tan K.M., Maxion R.A., Wing J.M // Carnegie Mellon University. – CMU-CS-07-146. – 2007.

40. *Шнайер Б.* Секреты и ложь. Безопасность данных в цифровом мире / Б. Шнайер. – СПб. : Питер, 2003. – 368 с.

41. *Полаженко С.* Деревья атак и их применение при анализе проблемы безопасности и защищённости программных продуктов [Электронный ресурс] / С. Полаженко. – Software-Testing.Ru. – 2008. – Режим доступа : http://software-testing.ru/library/testing/security/140-2008-10-02-18-15-03

42. *Sheyner O.* Automated Generation and Analysis of Attack Graphs / Sheyner O., Jha S., Wing J., Lippmann R., Haines J. // In 2002 IEEE Symposium on Security and Privacy. – Oakland, California. – 2002.

43. *Котенко И. В.* Интеллектуальная система анализа защищенности компьютерных сетей [Электронный ресурс] / И. В. Котенко, М. В. Степашкин, В. С. Богданов. – Режим доступа : http://www.positif.org/docs/SPIIRAS-NCAI'06-Stepashkin.pdf.

44. *Sheyner O.* Two Formal Analyses of Attack Graphs / O. Sheyner, S. Jha, J. Wing // IEEE Computer Security Foundations Workshop, Cape Brenton. – Nova Scotia, Canada. – June 2002. – P. 49–63.

45. *Колегов Д. Н.* Моделирование сетевых компьютерных систем с уязвимостями / Д. Н. Колегов // Прикладная дискретная математика. – 2009. – № 3. – С. 91–99.

46. *Wang Lingyu.* An Attack Graph-Based Probabilistic Security Metric / Lingyu Wang, Tania Islam, Tao Long, Anoop Singhal, Sushil Jajodia. – Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security. – London, 2008.

47. *Котенко И. В.* Оценка безопасности компьютерных сетей на основе графов атак и качественных метрик защищенности / И. В. Котенко, М. В. Степашкин, В. С. Богданов // Труды СПИИРАН. – Вып. 3, т. 2. – СПб. : Наука, 2006.

48. *Trost R.* Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century / R. Trost. – Addison Wesley Professional. – 2009. – 480 p.

49. *Саттер Г.* Стандарты программирования на C++ / Г. Саттер, А. Александреску; пер. с англ. – М. : Изд. дом «Вильямс», 2005. – 224 с. – ISBN 5-8459-0859-0 (рус.).

50. *Ховард М.* Как написать безопасный код на C++, Java, Perl, PHP, ASP.NET / М. Ховард, Д. Лебланк, Дж. Виега. – М. : ДМК Пресс, 2009. – 288 с.

51. *Гудлиф П.* Ремесло программиста. Практика написания хорошего кода / П. Гудлиф. – М. : Символ-Плюс, 2009. – 704 с.

52. *Seacord R. C.* Secure Coding in C and C++ / R. C. Seacord. – Addison-Wesley Professional. – 2005. – 368 p.

53. *Messier M.* Secure Programming Cookbook for C and C++: Recipes for Cryptography, Authentication, Input Validation & More / M. Messier. – O'Reilly Media. – 2003. – 792 p.

54. *Allen J. H.* Software Security Engineering: A Guide for Project Managers / J. H. Allen, S. Barnum. – Addison-Wesley Professional. – 2008. – 368 p.

55. *McGraw G.* Software Security: Building Security In / G. McGraw. – Addison-Wesley Professional. – 2006. – 448 p.

56. *Cave W. C.* Software Lifecycle Management: The Incremental Method / W. C. Cave, G. W. Maymon. – Pearson Higher Education. – 1984. – 300 p.

57. *Lee Y.* Integrating Software Lifecycle Process Standards with Security Engineering / Y. Lee, J. Lee, Z. Lee // Computers and Security. – v. 21, № 4. – 2002. – pp. 345–355.

58. *Howard M.* The Security Development Lifecycle Book / M. Howard, S. Lipner. – Microsoft Press. – 2006. – 352 p.

59. Open Web Application Security Project [Electron. resource]. – Access link : https://www.owasp.org

60. *Wysopal C.* The Art of Software Security Testing: Identifying Software Security Flaws / C. Wysopal, D. D. Zovi. – Addison-Wesley Professional. – 2006. – 312 p.

61. *Landauer L.* Hunting Security Bugs / L. Landauer. – Microsoft Press. – 2006. – 592 p.

62. *Walther B.* Web Security Testing Cookbook: Systematic Techniques to Find Problems Fast / B. Walther. – O'Reilly Media. – 2008. – 320 p.

63. *Thompson H. H.* How to Break Software Security / H. H. Thompson. – Addison Wesley. – 2003. – 208 p.

64. *Whittaker J. A.* How to Break Software: A Practical Guide to Testing / J. A. Whittaker. – Addison Wesley. – 2002. – 208 p.

65. *Microsoft*. Руководство по безопасности ОС Windows Server 2003 [Электронный ресурс]. – Режим доступа : http://technet.microsoft.com/ru-ru/library/cc163140.aspx

66. *Oracle*. Recommended Security Practices [Electron. resource]. – Access link : http://www.oracle.com/security/security-best-practices.html

67. *Cisco*. Service Provider Security Best Practices [Electron. resource]. – Access link : http://tools.cisco.com/security/center/serviceProviders.x?i=76

68. *Sun Microsysteems*. Sun blueprints online – Security [Electron. resource]. – Access link : http://www.sun.com/blueprints/browsesubject.html#security

69. *Convery S.* Network Security Architectures / S. Convery. – Cisco Press. – 2004. – 792 p.

70. *Kaeo M.* Designing Network Security / M. Kaeo. – Cisco Press. – 2003. – 768 p.

71. *Cole E.* Network Security Bible / E. Cole. – John Wiley & Sons. – 2009. – 936 p.

72. *Ретана А.* Принципы проектирования корпоративных IP-сетей / А. Ретана, Д. Слайс, Р. Уайт. – М. : Изд. дом «Вильямс», 2002. – 368 с.

73. *Биячуев Т.* Безопасность корпоративных сетей / Т. Биячуев; под ред. Л. Г. Осовецкого. – СПб. : СПб ГУ ИТМО, 2004. – 163 с.

74. *ISO Standards* – IT Security Techniques [Electron. resource]. – Access link : http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=45306

75. *ISO*/*IEC* 17799:2005. Information technology – Security techniques – Code of practice for information security management.

76. *Нормативно-правова база*. Стандарти [Електронний ресурс]. – Режим доступу : http://www.dstszi.gov.ua/dstszi/control/uk/publish/category?cat_id=38836

77. *ГОСТ* 51583-00. Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие требования.

78. *ГОСТ Р* 51624-00. Защита информации. Автоматизированные системы в защищенном исполнении. Общие требования.

79. *НД ТЗІ* 2.5-008-02. Вимоги із захисту конфіденційної інформації від несанкціонованого доступу під час оброблення в автоматизованих системах класу 2.

80. *НД ТЗІ* 2.5-010-03. Вимоги до захисту інформації Web-сторінки від несанкціонованого доступу.

81. *НД ТЗІ* 3.7-003-05. Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі.

82. *National* Security Agency. Defense in Depth: A practical strategy for achieving Information Assurance in today's highly networked environments [Electron. resource]. – Access link : http://www.nsa.gov/ia/_files/support/defenseindepth.pdf

83. *ГСТУ* СУІБ 2.0/ISO/IES 27002:2010. Інформаційні технології. Методи захисту. Звід правил для управління інформаційною безпекою.

84. *Грушо А. А.* Теоретические основы компьютерной безопасности / А. А. Грушо, Э. А. Применко, Е. Е. Тимонина. – М. : Академия, 2009. – 272 с.

85. *Астахов А.* Анализ защищенности корпоративных систем [Электронный ресурс] / А. Астахов. – 2002. – Режим доступа : http://www.osp.ru/os/2002/07-08/181720/_p1.html

86. *Pfleeger* Charles P. Security in Computing, 4th Edition / Charles P. Pfleeger, Shari Lawrence Pfleeger. – Prentice Hall. – 2006. – 880 p.

87. *Smith Sean*. The Craft of System Security / Sean Smith, John Marchesini. – Addison-Wesley Professional. – 2007. – 592 p.

88. *Астахов А. М.* Искусство управления информационными рисками / А. М. Астахов. – М. : ДМК Пресс, 2010. – 314 с.

89. *Positive* Technologies. Система контроля защищенности и соответствия стандартам MaxPatrol [Электронный ресурс]. – Режим доступа : http://www.ptsecurity.ru/download/maxpatrol1.pdf

90. *Брукс П.* Метрики для управления ИТ-услугами / П. Брукс. – М. : Альпина Бизнес Букс, 2008. – 288 с.

91. *Jaquith A.* Security Metrics: Replacing Fear, Uncertainty, and Doubt / A. Jaquith. – Addison-Wesley Professional. – 2007. – 336 p.

92. Jansen Wayne. Directions in Security Metrics Research [Electron. resource] / Wayne Jansen. – Computer Security Division Information Technology Laboratory National Institute of Standards and Technology. – 2009. – Access link : http://csrc.nist.gov/publications/nistir/ir7564/nistir-7564_metrics-research.pdf

93. *Berinato S.* A Few Good Information Security Metrics [Electron. resource] / S. Berinato. – CSO Online. – 2005. – Access link : http://www.csoonline.com/article/220462/ A_Few_Good_ Information_Security_Metrics

94. Securitymetrics.org. Metrics Definitions [Electron. resource]. – Access link : http://securitymetrics.org/content/Wiki.jsp?page=MetricsDefinitions

95. *Зегжда Д. П.* Основы безопасности информационных систем / Д. П. Зегжда, А. М. Ивашко. – М. : Горячая Линия – Телеком, 2000. – 452 с.

96. *Домарев В. В.* Безопасность информационных технологий. Системный подход / В. В. Домарев. – Киев : ООО ТИД «Диасофт», 2004. – 992 с.

97. *Карпов В. В.* Некоторые аспекты проблемы синтеза систем защиты информации от несанкционированного доступа / В. В. Карпов // Программные продукты и системы. – 2003. – № 3. – С. 10–11.

98. *Боня Ю. Ю.* Синтез системи захисту інформації, оптимальної за рівнем ризику / Ю. Ю. Боня, О. М. Новіков // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2007. – Вип. 1. – С. 26–33.

99. *Novikov A.* The synthesis of information protection systems with optimal properties. Complexity and Security. NATO Science for Peace and Security Series / A. Novikov, A. Rodionov. – E. Volume 37, 2008. – pp. 307–316.

100. *Задірака В. К.* Методи захисту банківської інформації / В. К. Задірака, О. С. Олексюк, М. О. Недашковський. – Київ : Вища шк., 1999. – 261 с.

101. *Можаев А. С.* Общий логико-вероятностный метод анализа надежности сложных систем : учеб. пособие / А. С. Можаев. – Л. : ВМА, 1988. – 68 с.

102. *Рябинин И. А.* Логико-вероятностные методы исследования надежности структурно-сложных систем / И. А. Рябинин, Г. Н. Черкесов. – М. : Радио и связь, 1981. – 264 с.

103. *Рябинин И.А.* Концепция логико-вероятностной теории безопасности / И. А. Рябинин // Приборы и системы управления. – 1993. – № 10. – С. 6–9.

104. *Медведовский И. Д.* Атака через Internet / И. Д. Медведовский, П. В. Семьянов, В. В. Платонов. – СПб. : НПО «Мир и Семья-95», 1997. – 296 с.

105. *Горфинкель С.* Безопасность Web и электронная коммерция / С. Горфинкель, Дж. Стеффорд. – М. : Book Media Publisher, 1999. – 458 с.

106. *Гайкович В.* Безопасность электронных банковских систем / В. Гайкович, А. Першин ; под ред. Ю. В. Гайковича. – М. : Изд-во «Единая Европа», 1993. – 368 с.

107. *Черней Г. А.* Оценка угроз безопасности автоматизированным информационным системам / Г. А. Черней // НТИ. Сер. 2, Информационные процессы и системы. – 1997. – № 10.

108. *Волик Б. Г.* Эффективность, надежность и живучесть управляющих систем / Б. Г. Волик, И. А. Рябинин // Автоматика и телемеханика. – 1984. – № 12. – С. 151–160.

109. *Ушаков И. А.* Методы исследования эффективности функционирования технических систем / И. А. Ушаков. – М. : Знание, 1976. – 64 с.

110. *De Finetti B.* Theory of Probability, Juhn Wiley and Sons, Inc. / De Finetti B. – New York, 1974.

111. *Apostolakis G.* Probabilistic Risk Assessment: The Subjectivistic Viewpoint and Some Suggestions, Nuclear Safety / G. Apostolakis. – 19 (3) June, 1978.

112. *Хенли Э.* Надежность технических систем и оценка риска / Э. Хенли, Х. Кумамото ; пер. с англ. – М. : Машиностроение, 1984. – 528 с.

113. *Risk* – Informed Inservice Inspection Evaluation Procedures. EPRITR-106706. – 1996.

114. *Bickel J.* Probabilistic Risk Assessment. PRA1-Concepts and Principles / J. Bickel, P. Moieni. –1996.

115. *Technical* Elements of Risk – Informed Inservice Inspection Programs for Piping. NUREG 1661. Draft Repoit. – 1999.

116. *Соложенцев Е. Д.* Моделирование поведения структурно-сложных систем: параметры, модели, знания, визуализация / Е. Д. Соложенцев, В. Е. Соложенцев ; под ред. И. А. Рябинина // Теория и информационная технология моделирования безопасности сложных систем. – Препринт 104. – СПб. : ИПМАШ РАН. – 1994. – Вып. 4. – С. 3–22.

117. *Королюк В. С.* Справочник по теории вероятностей и математической статистике / В. С. Королюк, Н. И. Портенко, А. В. Скороход, А. Ф. Турбин. – М. : Наука, 1985. – 640 с.

118. *Корн Г.* Справочник по математике для научных работников и инженеров / Г. Корн, Т. Корн. – М. : Наука ; Гл. ред. физ.-мат. лит., 1984. – 832 с.

119. *Гилл Ф.* Практическая оптимизация / Ф. Гилл, У. Мюррей, М. Райт. – М. : Мир, 1985. – 509 с.

120. *Реклейтис Г.* Оптимизация в технике: в 2 кн. / Г. Реклейтис, А. Рейвиндран, К. Рэгсдел ; пер. с англ. – М. : Мир, 1986.

121. *Финкельштейн М. С.* Надежность и живучесть радиоэлектрон ных систем / М. С. Финкельштейн. – М. : Отраслевая система НТИ, 1990. – 121 с.

122. *Волкович В. Л.* Модели и алгоритмы оптимизации надежности сложных систем / В. Л. Волкович, А. Ф. Волошин, В. А. Заславский, И. А. Ушаков ; под ред. В. С. Михалевича. – Киев : Наук. думка, 1992. – 312 с.

123. *Ушаков И.А.* Оценка, прогнозирование и обеспечение надежности сетей ЭВМ / И. А. Ушаков. – М. : Машиностроение, 1989. – 72 с.

124. *TrustWave*. Global Security Report 2010. – SpiderLabs. – 2010.

125. *Крис Касперски.* Техника сетевых атак [Электронный ресурс] / К. Касперски. – Режим доступа: https://www.trustwave.com/Resources/Library/Documents/2010-Trustwave-Global-Security-Report/

126. *Лукацкий А. В.* Выявление уязвимостей компьютерных сетей [Электронный ресурс] / А. В. Лукацкий. – Режим доступа : http://www.infocity.kiev.ua/hack/content/hack188.phtml

127. *Сычев А.* Анализ проблем защиты от внешнего нарушителя [Электронный ресурс] / А. Сычев, Д. Кузнецов. – Режим доступа : http://www.securitylab.ru/analytics/397848.php

128. *National* Vulnerability Database (NVD). CVE Statistics [Electron. resource]. – Access link : http://nvd.nist.gov/

129. *Рябинин И. А.* Логико-вероятностное исчисление монотонных и немонотонных функций алгебры логики [Электронный ресурс] / И. А. Рябинин. – Режим доступа : http://logic-cor.narod.ru/Ryabinin.pdf

130. *Рябинин И. А.* Феномен логико-вероятностного исчисления [Электронный ресурс] / И. А. Рябинин. – Режим доступа : http://www.szma.com/obzor3.pdf

131. *Рябинин И. А.* Логика и вероятность как инструменты исследования проблем надежности и безопасности сложных систем [Электронный ресурс] / И. А. Рябинин. – Режим доступа : http://www.szma.com/obzor6.pdf

132. *Рябинин И. А.* Надежность и безопасность структурно-сложных систем / И. А. Рябинин. – СПб. : Политехника, 2000. – 248 с.

133. *Панин О. А.* Анализ безопасности интегрированных систем защиты: Логико-вероятностный подход [Электронный ресурс] / О. А. Панин // Специальная техника. – 2004. – № 5. – Режим доступа : http://www.st.ess.ru /5-2004/

134. *Ершов Г. А.* Оценка безопасности атомных энергетических объектов на стадии проектирования / Г. А. Ершов, Ю. И. Козлов, А. С. Солодовников, А. С. Можаев // Тяжелое машиностроение. – 2004. – № 8. – С. 33–39.

135. *Виноградов И. М.* Математическая энциклопедия [Электронный ресурс] / И. М. Виноградов. – М. : Советская энциклопедия, 1977–1985. – Режим доступа : http://dic.academic.ru/dic.nsf/enc_mathematics/4086/

136. *Губко М. В.* Оптимальные иерархические структуры при монотонном функционале стоимости / М. В. Губко // Институт проблем управления им. В. А. Трапезникова РАН, Управление большими системами. – 2003. – № 3. – С. 27–34.

137. *Воронин А. А.* Оптимальные иерархические структуры / А. А. Воронин, С. П. Мишин. – М. : Ин-т проблем упр. им. В. А. Трапезникова РАН, 2003. – 214 с.

138. *Эйкхофф П.* Основы идентификации систем управления / П. Эйкхофф ; пер. с англ. – М. : Мир. – 1975. – 684 с.

139. *Соложенцев Е. Д.* Сценарное логико-вероятностное управление риском в бизнесе и технике / Е. Д. Соложенцев. – 2-е изд. – СПб. : Бизнес-пресса, 2006. – 560 с.

140. *Вишневский В. М.* Теоретические основы проектирования компьютерных сетей / В. М. Вишневский. – М. : Техносфера, 2003. – 512 с.

141. *Motter A. E.* Introduction: Optimization in networks / A. E. Motter, Z. Toroczkai // Chaos. –2007. – Volume 17, Issue 2. – pp. 026101-026101-3.

142. *Зайченко Ю. П.* Компьютерные сети / Е. Ю. Зайченко, Ю. П. Зайченко. – Киев : Слово, 2010. – 520 с.

143. *Convery* Sean. Network Security Architectures / Sean Convery. – Cisco Press. – 2004. – 792 p.

144. *Cisco*. Cisco Validated Design Program [Electron. resource]. – Access link : http://www.cisco.com/en/US/netsol/ ns741/networking_solutions_program_home.html

145. *Microsoft*. Microsoft Systems Architecture [Electron. resource]. – Access link : http://www.microsoft.com/Rus/Business/Vision/Strategy/Architecture.mspx

146. *Cisco*. Internetwork Design Guide [Electron. resource]. – Access link : http://www.cisco.com/en/US/docs/internetworking/design/guide/idg4.html

147. *Esposito M.* An efficient approach to the network division problem, VLANs configuration and WLANs hosts grouping / M. Esposito, A. Pescapè, G. Ventre // 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2004). – San Francisco Bay Area (USA). – 2004. – pp. 241–246.

148. *Rooney S.* Automatic VLAN Creation Based on On-line Measurement / S. Rooney, C. Hortnagl, J. Krause // ACM SIGCOMM Computer Communication Review. – 1999. – 29 (3). – pp. 50–57.

149. *Зайченко Ю. П.* Дослідження операцій / Ю. П. Зайченко. – Київ : Слово, 2001. – 688 с.

150. *Сергиенко И. В.* Математические модели и методы решения задач дискретной оптимизации / И. В. Сергиенко. – Киев : Наук. думка, 1988. – 472 с.

151. *Михалевич В. С.* Методы последовательной оптимизации в дискретных сетевых задачах оптимального распределения ресурсов / В. С. Михалевич, А. И. Кукса. – М. : Наука, 1983. – 208 с.

152. *Ковалев М. М.* Дискретная оптимизация. Целочисленное программирование / М. М. Ковалев. – М. : Едиториал УРСС, 2003. – 192 с.

153. *Соложенцев Е. Д., Соложенцев В. Е.* Концепции обеспечения безопасности сложных систем: «нулевого риска», «ненулевого риска», смешанного подхода / Е. Д. Соложенцев, В. Е. Соложенцев ; под ред. И. А. Рябинина, Е. Д. Соложенцева. – Вып. 4. – СПб. : ИПМАШ РАН, 1994. – Препринт 110. – С. 67–82.

154. *ISO/IEC* 2nd WD 13335-3, Information technology – Security techniques – Guidelines for the management of information and communications technology security – Part 3: Techniques for information and communications technology security risk management.

155. *Балаш Э.* Аддитивный алгоритм для решения задач линейного программирования с переменными, принимающими значения 0 или 1 / Э. Балаш // Кибернетич. сб., Новая серия. – Вып. 6. – 1969.

156. *Geoffrion A.M.* Integer Programming by Implicit Enumeration and Balas' Method / A. M. Geoffrion // SIAM Review. – 1967. – Vol. 9, No. 2 (April). – pp. 178–190.

157. *Михалевич В. С.* Вычислительные методы исследования и проектирования сложных систем / В. С. Михалевич, В. Л. Волкович. – М. : Наука, 1982.

158. *Артеменко В. І.* Про метод розв'язування задач линійного програмування з бульовими змінними / В. І. Артеменко, І. В. Сергієнко // Доповіді АН УРСР № 4, серія А. – 1980.

159. *Емеличев В. А.* Метод построения последовательности планов для решения задач дискретной оптимизации / В. А. Емеличев, В. И. Комлик. – М. : Наука, 1981. – 208 с.

160. *Geoffrion Arthur M.* Integer Programming by Implicit Enumeration and Balas' Method / Arthur M. Geoffrion // SIAM Review. – Vol. 9, No. 2 (Apr., 1967). – p. 178–190.

161. *Сергиенко И. В.* Приближенные методы решения дискретных задач оптимизации / И. В. Сергиенко, Т. Т. Лебедева, В. А. Рощин. – Киев : Наук. думка. – 1980. – 276 с.

162. *Систематический* каталог библиотеки численного анализа НИВЦ МГУ [Электронный ресурс]. – Режим доступа : http://srcc.msu.su/num_anal/lib_na/cat/cat0.htm

163. *ГОСТ* 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

164. *Олифер В. Г.* Стратегическое планирование сетей масштаба предприятия / В. Г. Олифер, Н. А. Олифер. – М. : Центр Информ. Технологий. – 2000. – 680 с.

165. *Леохин Ю. Л.* Корпоративные сети – транспортная система ИТ инфраструктуры предприятий / Ю. Л. Леохин // Качество. Инновации. Образование. – 2009. – № 1. – С. 45–50.

166. *Новіков О. М.* Логіко-функціональні моделі безпеки інформації в інформаційно-обчислювальних системах з відкритою архітектурою / О. М. Новіков, А. О. Тимошенко // Наукові вісті Національного технічного університету України «Київський політехнічний інститут». – 2002. – № 2. – С. 40–46.

167. *Боня Ю. Ю.* Синтез оптимальної системи захисту інформації на основі методу вектора спаду / Ю. Ю. Боня, О. М. Новіков // Наукові вісті НТУУ «КПІ». – 2006. – № 4. – С. 107–112.

168. *Боня Ю. Ю.* Алгоритми синтезу оптимальної системи захисту інформації / Ю. Ю. Боня, О. М. Новіков // Системні дослідження та інформаційні технології. – 2007. – № 3. – С. 68–81.

ДЛЯ НОТАТОК

Навчальне видання

Новіков Олексій Миколайович
Родіонов Андрій Миколайович
Тимошенко Андрій Олександрович

Моделі та методи кібернетичного захисту інформаційно-комунікаційних систем на основі логіко-ймовірнісного підходу

Монографія

(Російською мовою)

Коректура *Н. В. Мурашової, Л. О. Коротіної*
Верстання *С. А. Боброва, А. М. Корнієнка*

Темплан 2015 р., поз. 3-1-019

Підп. До друку 21.01.2016. Фориат 60х84 $^1/_{16}$. Папір офс. Гарнітура Times.
Спосіб друку – ризографія. Ум. друк. арк.. 16,04. Обл.-вид. арк.. 26,68. Зам. №15-235.
Наклад 300 пр.

Національний технічний університет України
«Київський політехнічний інститут»
Видавництво «Політехніка»
Свідоцтво ДК № 1665 від 28.01.2004 р.
03056, м. Київ, вул.. Політехнічна, 14, корп. 15
тел. (44) 204-81-78