МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ»**

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ПРОГРАММИРОВАНИЯ

ЗАЩИТА ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

***Под редакцией д-ра экон. наук Е.В. Стельмашонок,
канд. физ.-мат. наук И.Н. Васильевой***

**ИЗДАТЕЛЬСТВО
САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО
ЭКОНОМИЧЕСКОГО УНИВЕРСИТЕТА
2017**

ББК 32.971.35-5
З40

З40 **Защита** информации в компьютерных системах / под ред. д-ра экон. наук Е.В. Стельмашонок, канд. физ.-мат. наук И.Н. Васильевой. – СПб. : Изд-во СПбГЭУ, 2017. – 163 с.

ISBN 978-5-7310-4070-9

Монография посвящена проблемам моделирования и построения защищенных компьютерных систем. Исследуются вопросы информационной безопасности при применении современных информационных технологий в различных сферах профессиональной деятельности. Рассматриваются методы и средства защиты информации, модели управления информационной безопасностью, приводится анализ информационных рисков и оценка эффективности систем защиты информации.

Монография может быть полезна студентам, магистрантам, аспирантам, которые занимаются исследованием данных проблем и специалистам в области защиты информации, а также ИТ специалистам и менеджерам компаний, всем, кто интересуется применением новых информационных технологий и связанными вопросами информационной безопасности.

The problems of modeling and building of protected computer systems are described. The issues of information security in the application of modern information technologies in various spheres of professional activity are investigated. Methods and means of information protection, information security management models are considered, information security risks are analyzed and the effectiveness of information security systems is evaluated.

The publication can be useful to students, undergraduates, graduate students who research these problems, to professionals in the field of information security, as well as IT professionals and company managers, all who are interested in use of new information technologies and the connected questions of information security.

ББК 32.971.35-5

Рецензенты: засл. деят. науки РФ, д-р техн. наук, проф. **В.В. Трофимов** (СПбГЭУ)
д-р техн. наук, проф. **А.Ю. Иванов** (СПб ун-т МВД России)

ISBN 978-5-7310-4070-9

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В настоящее время число угроз информационной безопасности непрерывно растет, они становятся все более разнообразными, а их реализация наносит значительный ущерб как организациям, так и частным лицам. Такая ситуация обусловливает актуальность решения проблем обеспечения информационной безопасности и защиты информации. Эти вопросы неразрывно связаны с теорией и практикой применения современных информационных технологий. Модель угроз информационной безопасности и требования к защите информации проектируемой или модифицируемой информационной системы, на основании которых производится выбор необходимого уровня защищенности системы, определяются на этапе формирования технического задания. Эти требования во многом определяют выбор используемых технологий и подходов к построению компьютерных систем. Должны быть разработаны соответствующие теоретические основы и модели с учетом актуальных угроз и положений информационной безопасности. Особого внимания заслуживает решение проблем защиты персональных данных при использовании сетевых сервисов, поддерживающих решения в области цифровой экономики и интернета вещей.

Общие вопросы построения и безопасности информационных систем рассмотрены в первой главе монографии. Требования к защите информации определяют состав, функции и особенности реализации подсистемы защиты информации и должны выполняться, контролироваться и актуализироваться на всех этапах жизненного цикла компьютерной системы.

Вторая глава монографии посвящена описанию отдельных методов и средств защиты информации. Подробно рассмотрены вопросы практического использования криптографических протоколов и криптографических методов защиты, уделено внимание анализу и разработке систем обнаружения вторжений и поиску уязвимостей в программном коде

Вместе с тем, все более пристальное внимание уделяется рациональному выбору состава системы защиты информации и его обоснованию на основе критериев экономической эффективности, анализу рисков и управлению информационной безопасностью. Эти проблемы предлагается решать на основе категорирования информационных активов при формировании требований к защите информации и методов математического моделирования, что и отражено в третьей главе монографии.

Текст монографии подготовлен коллективом авторов, вклад каждого из авторов указан ниже.

- Буйневич М. В. – доктор техн. наук, профессор, профессор Санкт-Петербургского государственного экономического университета, п. 2.3.

- Васильева И. Н. – канд. физ.-мат. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, доцент Санкт-Петербургского университета МВД РФ, п. 2.1.
- Воробьев Т. М. – студент направления «Прикладная математика и информатика» Санкт-Петербургского государственного экономического университета, п. 1.3.
- Гниденко И. Г. – канд. экон. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, п. 1.1.
- Егорова И. В. – канд. экон. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, п. 1.1.
- Еникеева Л. А. – доктор экон. наук, профессор, профессор Санкт-Петербургского государственного института кино и телевидения, п. 1.5.
- Зельман С. Г. – студент направления «Информационная безопасность» Санкт-Петербургского государственного экономического университета, п. 3.4.
- Израилов К. Е. – канд. техн. наук, старший научный сотрудник ЗАО «Фирма «Пассат» (г. Санкт-Петербург), п. 2.3.
- Ишанханов С. Р. – магистрант Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики, п. 2.2.
- Куватов В. И. – доктор техн. наук, профессор, засл. деятель науки РФ, профессор Санкт-Петербургского университета МВД РФ, п. 3.1.
- Локтионов О. В. – начальник центра информационных технологий, связи и защиты информации Главного управления МВД РФ по Санкт-Петербургу и Ленинградской области, п. 1.7.
- Малахова П. А. – магистрант Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики, п. 3.5.
- Мердина О. Д. – канд. экон. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, п. 3.4.
- Полегенько А. М. – старший преподаватель Санкт-Петербургского государственного экономического университета, аспирант Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики, специалист по защите информации ЗАО «ТЕЛРОС» (г. Санкт-Петербург), п. 1.6.
- Попов М. А. – студент направления «Информационная безопасность» Санкт-Петербургского государственного экономического университета, п. 1.2.
- Примакин А. И. – доктор техн. наук, профессор, профессор Санкт-Петербургского университета МВД РФ, п. 3.1.

- Семенова Т. Г. – Microsoft certified professional, менеджер проектов SAM компании CSI Group (г. Москва), п. 1.8.
- Семенова С. О. – аспирант Санкт-Петербургского государственного экономического университета, специалист по защите информации ЗАО «ТЕЛРОС» (г. Санкт-Петербург), пп. 1.8, 3.3.
- Соловьев А. И. – доктор полит. наук, профессор, профессор Финансового университета при Правительстве Российской Федерации (г. Москва), п. 3.6.
- Соловьев С. А. – канд. экон. наук, доцент, доцент Финансового университета при Правительстве Российской Федерации (г. Москва), п. 3.6.
- Солодянников А. В. – канд. техн. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, генеральный директор ЗАО «Ассоциация специалистов информационных систем» (ЗАО «АСИС», г. Санкт-Петербург), п. 3.8.
- Соколов Р. В. – доктор экон. наук, профессор, профессор Санкт-Петербургского государственного экономического университета, п. 3.2.
- Соколовская С. А. – канд. экон. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, п. 1.4.
- Стельмашонок В. Л. – канд. экон. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, п. 3.7.
- Стельмашонок Е. В. – доктор экон. наук, профессор, профессор Санкт-Петербургского государственного экономического университета, п. 3.7.
- Торосян Е. К. – канд. экон. наук, доцент Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики, п. 1.5.
- Федоров Д. Ю. – старший преподаватель Санкт-Петербургского государственного экономического университета, пп. 1.2, 1.3.
- Чернокнижный Г. М. – канд. техн. наук, доцент, доцент Санкт-Петербургского государственного экономического университета, пп. 2.2, 3.5.

ГЛАВА 1. ОБЩИЕ ВОПРОСЫ ПОСТРОЕНИЯ И БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1. О современных проблемах информационной безопасности

Гниденко И.Г., Егорова И.В.

С широким распространением применения мобильных устройств и роста популярности сетевых сервисов возник целый ряд новых проблем и угроз компьютерной безопасности. При этом часть проблем относится к безопасности корпоративных и служебных данных, а другая угрожает безопасности личных данных обычных пользователей.

Стало трудно полностью отделить личные данные от служебных. Смартфоны, которые являются личными устройствами, могут подключаться к корпоративным сетям и одновременно использоваться для ведения частной переписки и доступа в социальные сети. Это создает угрозу, как целостности, так и обеспечению конфиденциальности корпоративных или служебных данных. И здесь по-прежнему лучшим решением может являться запрет использования личных устройств или ограничения для сотрудников на использование некоторых сетевых ресурсов, например, социальных сетей.

Угрозы рядовым пользователям можно разделить на несколько категорий: угрозы, связанные с похищением персональных данных; угрозы целостности данных; угрозы, связанные с несанкционированным использованием принадлежащих пользователю компьютерных устройств и нарушением их работы.

Широкое распространение интернет магазинов и электронных платежных систем, создает угрозу похищения персональных данных пользователя. Утрата смартфона и пренебрежение элементарными правилами обеспечения безопасности могут привести к пропаже денег со счетов и получению злоумышленниками кредитов от имени пользователя [6].

Сетевые сервисы, такие как: электронная почта, социальные сети, интернет магазины, облачные хранилища данных, платежные системы, требуют от пользователей предоставления личных данных в различном объеме. При этом личные данные могут быть похищены как при атаках на сервисы из сети интернет, так и недобросовестными сотрудниками, обеспечивающими работу этих сервисов. После этого персональные данные могут быть использованы преступниками при совершении мошеннических сделок [4, 5]. Правда, для этого обычно нужны не просто персональные данные, а сканированные паспортные данные.

Ситуацию усугубляет обнаружение серьезных ошибок в системном программном обеспечении, а такие ошибки выявляются регулярно. Последний инцидент связан с обнаружением критической уязвимости в протоколе WPA2, который используется для защиты передаваемых данных в сетях Wi-Fi.

Обнаруженная уязвимость позволяет читать зашифрованную информацию, передаваемую между точкой доступа и пользовательскими устройствами. Информация об уязвимости была опубликована в октябре 2017 года, но исправления некоторыми производителями оборудования и программного обеспечения будут выпущены только в начале 2018 года [7]. И это обычная для таких случаев ситуация, выпуск исправлений требует от нескольких дней до нескольких месяцев.

Недостаточные знания рядовых пользователей о компьютерной безопасности также ухудшают ситуацию, не все пользователи понимают необходимость применения антивирусов [3]. По данным собранным Microsoft более чем на 50% компьютеров с операционной системой Windows 10 антивирус либо выключен, либо не обновляется [2]. Не лучше ситуация и на мобильных устройствах под управлением Android.

В последнее время и некоторые специалисты по компьютерной безопасности говорят о бесполезности антивирусного программного обеспечения. Это объясняется тем, что даже использование антивируса не гарантирует полную защиту компьютера или мобильного устройства, так как постоянно появляются новые угрозы и способы проникновения вредоносного программного обеспечения на компьютеры, планшеты и смартфоны.

Кроме того, сами антивирусы несут потенциальную угрозу для пользователей, так как для своей работы требуют прав на доступ к ресурсам компьютера, которых обычно не имеет рядовой пользователь. Поэтому антивирус может собирать данные и отправлять их в указанное место. Именно поэтому в сентябре 2017 года Сенат Конгресса США запретил использование продуктов «Лаборатории Касперского» государственными учреждениями. Впрочем, «Лаборатория Касперского» предлагает открыть код своих продуктов для независимых экспертов, чтобы все заинтересованные могли убедиться, что никаких действий кроме борьбы с вирусами эти продукты не выполняют.

Вероятно, аналогичный запрет в отношении антивирусных средств иностранного производства в ближайшее время будет введен в нашей стране.

Кроме применения антивирусов имеется еще ряд рекомендаций для пользователей, которые позволяют повысить их уровень безопасности. Список таких рекомендаций можно найти на сайтах ведущих отечественных производителей антивирусов: «Лаборатория Касперского», «Доктор Веб». Для пользователей банковских систем рекомендации по безопасности обычно приведены на сайте соответствующего банка.

Одной из рекомендуемых мер безопасности является применение только легального программного обеспечения. Однако известны случаи распространения троянских программ вместе с программным обеспечением, полученным из официального магазина GooglePlay. В сентябре 2017 года в игре

«Jewels Star Classic» была обнаружена функциональность, предназначенная для похищения данных банковских карт [8].

Через некоторое время после установки игра запрашивала разрешения на доступ к специальным возможностям Android, если пользователь давал доступ, то программа могла выводить поддельное окно для настройки платежного сервиса. При этом запрашивались данные банковской карты, которые потом отправлялись вирусописателям.

Этот пример показывает важность следования еще одной рекомендации для пользователей Android: при установке программы и в дальнейшем при ее эксплуатации внимательно относится к запрашиваемым программой разрешениям. Возможно, иногда лучше отказаться от пользования программой, если не ясно с какой целью она запрашивает некоторые разрешения.

К угрозам целостности данных относятся некоторые вирусы, блокирующие работу компьютера. При этом они могут зашифровывать данные пользователя. За разблокировку и расшифровку требуют вознаграждение. При этом даже в случае уплаты вознаграждения данные не всегда восстанавливаются. Чтобы не потерять важную информацию пользователям рекомендуется регулярно выполнять резервное копирование.

Способом распространения вирусов-вымогателей может быть, например, письмо, полученное по электронной почте. Поэтому еще одна рекомендация по безопасности для пользователей заключается в том, чтобы не открывать письма, полученные из неизвестных источников, такие письма лучше сразу удалять.

Отчасти проблема нежелательной и опасной почты решается применением спам-фильтров, на сайтах почтовых систем. Подозрительные письма при этом помещаются в специальную папку. Спам-фильтры опираются при фильтрации на постоянно пополняемую и обновляемую базу правил отбраковки подозрительных писем. При чтении писем из такой папки через браузер на сайте почтовой системы некоторые действия могут блокироваться, например, могут не загружаться изображения.

Защитить данные пользователя от несанкционированного доступа позволяют системы идентификации и аутентификации. К ним, прежде всего, относятся системы парольной защиты. Широкое распространение таких систем обусловлено низкой стоимостью и простотой организации и использования.

Однако надежность парольных систем защиты во многом ограничивается действиями самих пользователей, которые предпочитаю использовать несложные пароли, одинаковые для различных ресурсов, а также халатно относятся к хранению паролей.

Исследования, проводимые специалистами в области информационной безопасности, позволило выявить наиболее часто используемые пароли [1] (см. табл. 1.1).

Таблица 1.1

Наиболее часто используемые пароли

Пароль	Доля (%)
1234567	3,36
12345678	1,65
123456	1,02
Пустая строка	0,72
12345	0,47
7654321	0,31
qweasd	0,27
123	0,25
qwerty	0,25
123456789	0,23

Для раскрытия пароля пользователя может быть использован метод тотального перебора, при котором подбор пароля осуществляется с помощью проверки всех возможных сочетаний символов. Сложность и длительность такого подбора существенно возрастает с увеличением длины пароля, а также мощности алфавита (количеством различных символов, используемых в пароле).

Многие пользователи предпочитают использовать короткие, легко запоминаемые пароли, содержащие небольшой набор символов [1] (см. табл. 1.2, 1.3).

Таблица 1.2

Наиболее часто используемые наборы символов

Набор символов	Доля (%)
Только цифры	52,73
Символы английского алфавита в нижнем регистре	17,96
Символы английского алфавита в нижнем регистре и цифры	17,51
Символы английского алфавита в разных регистрах и цифры	3,4
Символы английского алфавита в разных регистрах	1,63
Символы английского алфавита в верхнем регистре и цифры	1,35
Символы русского алфавита в нижнем регистре	1,12

Таблица 1.3

Длина используемых пользователем паролей

Количество символов	Доля (%)
0	0,71
1	0,26
2	0,39
3	1,37
4	2,03
5	4,86
6	27,22
7	21,75
8	25,22
9	6,5
10	4,42
11	2,83
12	1,33
13	0,4
14	0,34
15	0,1
16	0,09
17	0,02
18	0,01
19	0,01
20	0,008
>20	0,02

Разработчики современных информационных ресурсов требует от пользователя задания длинных паролей, обязательно включающих как прописные и строчные буквы русского и латинского алфавита, так и цифры, и специальные символы.

Помимо простого перебора для раскрытия пароля пользователя может использоваться метод словарной атаки, при котором в качестве паролей проверяются слова, входящие в публично распространяемые словари, или их модификация (изменение раскладки клавиатуры, порядка следования букв в слове на обратный и т.д.).

Наконец, для раскрытия пароля может быть использована личная информация о пользователе (имена и фамилии пользователя и членов его семьи, даты рождения, клички домашних животных и т.д.). Статистика использования подобных «слабых» с точки зрения возможности раскрытия паролей [1] приведена в таблице 1.4.

Таблица 1.4

Статистика использования легко раскрываемых паролей

Содержание пароля	Доля (%)
Полное совпадение пароля с именем пользователя	3,94
Частичное совпадение пароля с именем пользователя	0,7
Пароль содержится в публично распространяемых словарях	14,69
Пароль является пустой строкой	0,7

Кроме проблемы выбора стойкого к раскрытию пароля существует также проблема его надежного хранения. Часто пользователи не уделяют должного внимания обеспечению секретности своих паролей. Пароли могут храниться в файлах на дисках или даже на бумажках, приклеенных к блокам компьютера. Одни и те же пароли могут использоваться в течение длительного периода, их своевременная смена не производится. Более того, одни и те же пароля могут использоваться для доступа к различным ресурсам.

Все эти факторы существенно снижают эффективность парольной защиты.

К угрозам третьего типа можно отнести троянские программы, которые позволяют удаленно управлять зараженным компьютером и пользоваться им для организации хакерских атак или для рассылки спама.

Появление крипто-валюты Bitcoin, которую можно получить путем генерации на компьютере, породило новый вид угроз. Для создания крипто-валюты нужны большие вычислительные мощности, которые можно получить, заразив вирусом чужие компьютеры и запустив на них генерацию крипто-валюты. Можно даже не использовать вирус, а встраивать скрипт для получения крипто-валюты в код сайта. Обращение к такому сайту приведет к запуску скрипта на устройстве пользователя для получения крипто-валюты. С этим можно было бы смириться, но наличие открытой страницы такого сайта в браузере приводит к практически полной загрузке процессора и замедляет выполнение любой задачи, практически блокируя работу.

Можно сделать вывод, что число угроз информационной безопасности растет, и они становятся более разнообразными, а методы противодействия им более сложными и нет универсального средства, применение которого гарантировало бы полную безопасность.

Минимальными требованиями к защите персональных устройств являются: применение антивируса, использование программ и данных, полученных только из надежных легальных источников, выполнение резервного копирования наиболее важной информации. Но более защищенными будут

пользователи, знающие об актуальных угрозах и понимающие механизмы, которыми пользуются злоумышленники. Поэтому актуальной является задача обучения пользователей основам компьютерной безопасности.

Литература:

1. Анализ проблем парольной защиты в российских компаниях [электронный ресурс]. URL: http://www.securitylab.ru/analytics/381943.php (дата обращения 11.11.2017).
2. Антивирусная правДА! Желтый уровень безопасности [электронный ресурс]. URL: https://www.drweb.ru/pravda/issue/?number=397 (дата обращения 11.11.2017).
3. Васильева И.Н. Управление информационной безопасностью. Учебное пособие. – СПб: Изд-во СПбГЭУ. 2014. – 172 с.
4. Гниденко И.Г., Мердина О.Д. Методы защиты программного обеспечения от несанкционированного доступа // Международная научно-практическая конференция «Перспективы развития науки и образования». Сб.науч.тр. – М.: АР-Консалт, 2013. – С. 110-115.
5. Как берутся кредиты по чужому паспорту [электронный ресурс]. URL: http://www.securitylab.ru/blog/company/securityinform/152810.php (дата обращения 11.11.2017).
6. Как персональные данные могут использоваться для мошенничества с сотовыми операторами [электронный ресурс]. URL: http://www.securitylab.ru/blog/company/securityinform/153190.php (дата обращения 11.11.2017).
7. Опубликована подробная информация о проблемах WPA2 [электронный ресурс]. URL: https://xakep.ru/2017/10/16/wpa2-krack-2/ (дата обращения 11.11.2017).
8. Троянец в Google Play [электронный ресурс]. URL: https://news.drweb.ru/show/review/?lng=kk&i=11505#googleplay (дата обращения 11.11.2017).

1.2. Пример построения онтологии в области защиты информации

Федоров Д. Ю., Попов М.А.

Обеспечение государственно-правового регулирования предполагает опору на формализованную нормативно-правовую базу. Это относится к любой области, в частности, к защите информации. Одним из способов формализации является построение онтологии в области защиты информации. Онтология – это попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы. Онтологии используются как форма представления знаний о реальном мире или его части [1].

Онтология может быть использована как источник энциклопедических знаний в области защиты информации, например, гораздо проще «заполнять пробелы» в знаниях или изучать новые области через графическое представление. С ее помощью можно определять, к какой сфере защиты информации относится то или иное понятие. Это может быть полезно, например, для анализа текстов, для проверки актуальности определения (насколько давно введено понятие).

Так как в стандартах описаны, в том числе, процессы обеспечения информационной безопасности, анализ онтологии может позволить проверить правильность существующего процесса или спланировать новый.

Рассмотрим алгоритм формирования (этапы построения) онтологии для области защиты информации.

1. Создается класс, совпадающий с названием нормативного документа (далее – НД).
2. Описание и область применения НД заносятся в комментарий к классу.
3. НД логически разделяются и создаются подклассы.
4. Для подклассов создаются другие подклассы до тех пор, пока полностью не будет раскрыто содержание НД.
5. Определения заносятся в комментарий подкласса.
6. Примечания с определениями заносятся в комментарии.

Изначально был рассмотрен ГОСТ Р 50922–2006, содержащий основные термины, использующиеся в защите информации. В отдельный класс были выделены определения общетехнических понятий. Следующим в онтологию был внесен ГОСТ Р 50.1.053–2005, так как в нем содержатся термины, относящиеся к технической защите информации. Также в отдельный класс были вынесены иноязычные эквиваленты терминов. Далее был создан подкласс «Методы проведения программных средств на наличие вирусов», в комментарии была занесена основная информация раздела, к этому классу был создан подкласс «Программные методы», к которому были созданы подклассы, перечисляющие методы. Информация о том, чем отличаются программно-аппаратные методы, была занесена в комментарий к классу «Программные методы». Определения программных методов были занесены в комментарии к соответствующим классам.

Затем был создан подкласс «Порядок проведения испытаний программных средств», основные положения пункта были занесены в комментарии. К нему был создан подкласс «Меры по защите проверяемых программных средств», к этому классу были созданы подклассы: «Меры по защите проверяемых программных средств», «Технические средства испытательного стенда», «Обязанности испытательного стенда», «Состав работ по подготовке» и «Проверка программных средств на наличие

компьютерных вирусов». Данный ГОСТ очень сильно устарел, было даже несколько неловко заносить информацию о том, например, что информацию необходимо передавать на дискетах. Попытки найти более новую версию ГОСТа успехом не увенчались.

В стандарте ГОСТ Р 51275–2006 «Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения» первым подклассом были введены «Термины и определения» по аналогии со всеми предыдущими, далее были созданы подклассы «Основные положения» и «Классификация факторов». Основные положения были занесены в комментарии. В классификации основная информация также была в комментариях, но факторы были вынесены в подкласс.

По аналогии в онтологию были внесены ГОСТ Р 50739–95 «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования», ГОСТ Р 51898–2002 «Аспекты безопасности. Правила включения в стандарты», ГОСТ Р 52069.0-2003 «Защита информации. Система стандартов. Основные положения», ГОСТ Р 52447–2005 «Защита информации. Техника защиты информации. Номенклатура показателей качества», ГОСТ Р ИСО/МЭК 15026–2002 «Информационная технология. Уровни целостности систем и программных средств», ГОСТ Р ИСО/МЭК 17799–2005 «Информационная технология. Практические правила управления информационной безопасностью». В процессе работы часто было трудно найти определенный стандарт или его переизданную версию.

На рисунке 1.1 представлен фрагмент онтологии для области защиты информации.

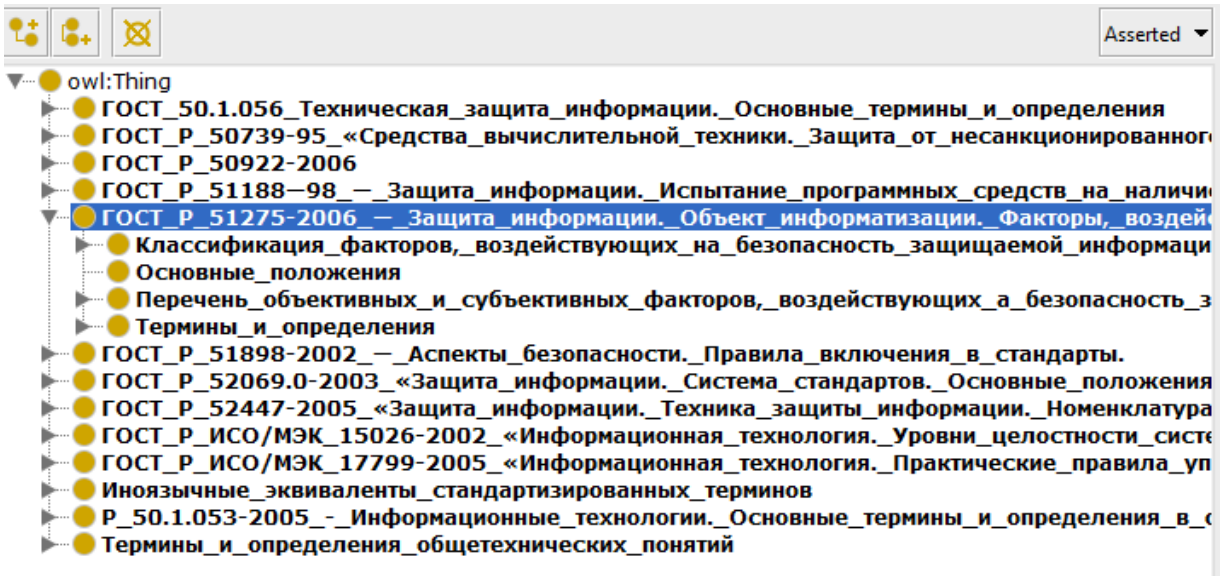


Рис. 1.1. Фрагмент онтологии для области защиты информации

На рисунке 1.2 представлен фрагмент визуального представления онтологии для области защиты информации.

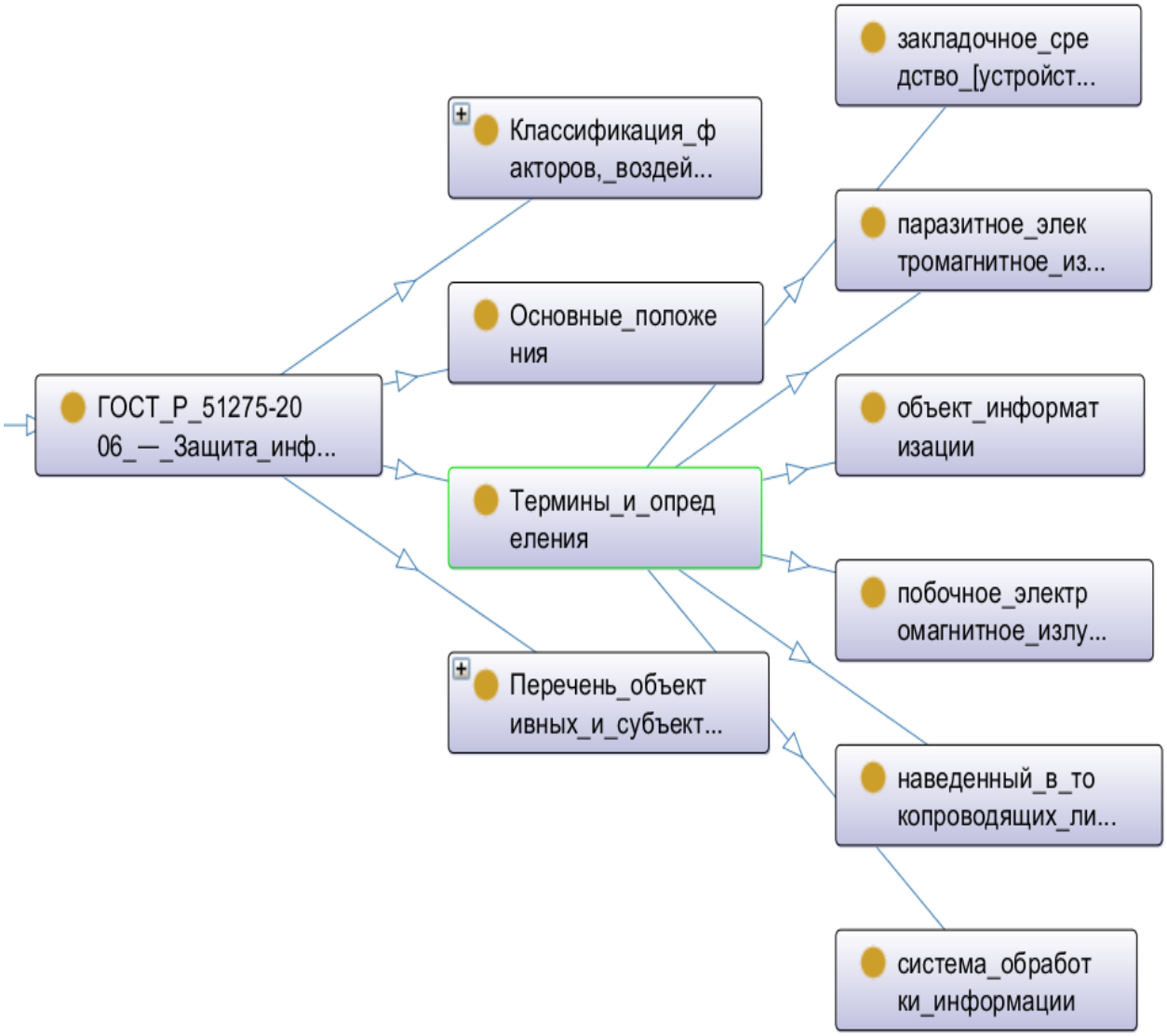


Рис. 1.2. Фрагмент визуального представления онтологии
для области защиты информации

Для демонстрации автоматизированного анализа полученной онтологии был выбран язык Python [2] и модуль Owlready2[1]. В результате была разработана программа, позволяющая через анализ онтологии получить определение термина и соответствующие ему предыдущий и последующий субклассы.

На рисунке 1.3 представлена блок схема алгоритма работы программы.

[1] Полная документация находится по адресу: http://pythonhosted.org/Owlready2/

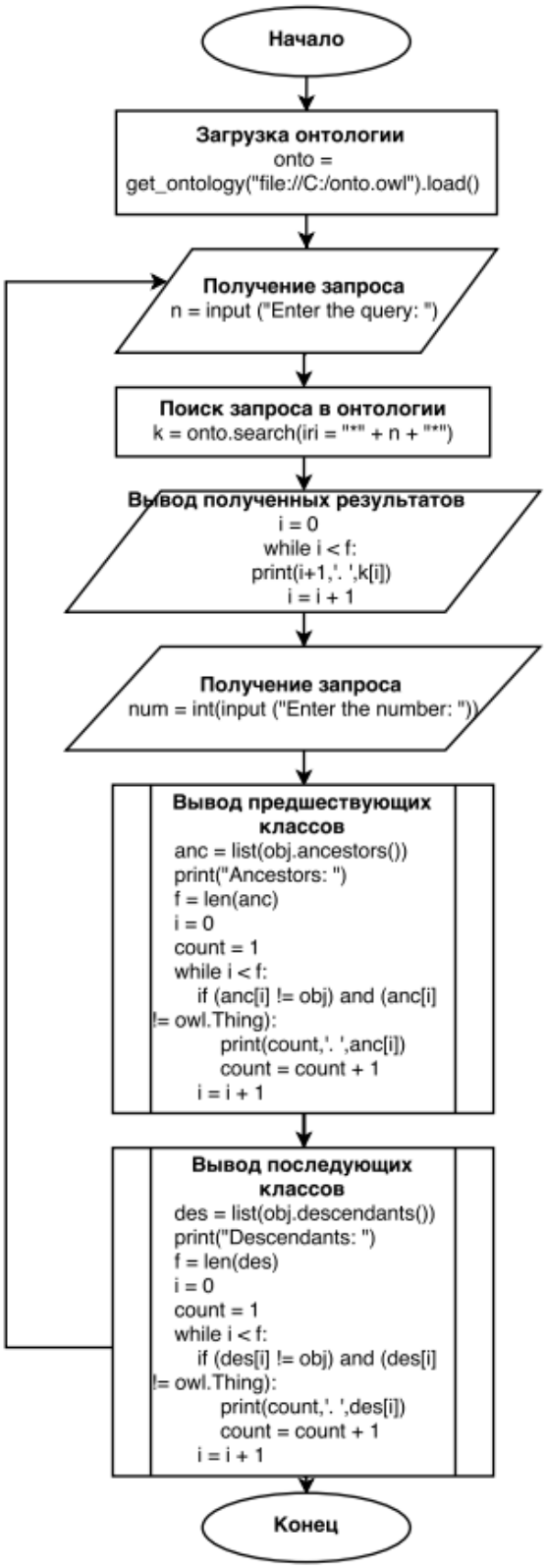


Рис. 1.3. Блок схема алгоритма работы программы

Далее представлен исходный текст программы:

```
from owlready2 import*
# Загрузка онтологии
```

```
onto = get_ontology("file://C:/onto.owl").load()
while True:
    n = input ("Enter the query: ")
    k = onto.search(iri = "*" + n + "*")
    f = len(k)
    i = 0
    while i < f:
        print(i+1,'. ',k[i])
        i = i + 1
    num = int(input ("Enter the number: "))
    obj = k[num-1]
    if obj.comment:
        print(obj, " - ", obj.comment)
    # предыдущий субкласс:
    anc = list(obj.ancestors())
    print("Ancestors: ")
    f = len(anc)
    i = 0
    count = 1
    while i < f:
        if (anc[i] != obj) and (anc[i] != owl.Thing):
            print(count,'. ',anc[i])
            count = count + 1
        i = i + 1
    # последующий субкласс:
    des = list(obj.descendants())
    print("Descendants: ")
    f = len(des)
    i = 0
    count = 1
    while i < f:
        if (des[i] != obj) and (des[i] != owl.Thing):
            print(count,'. ',des[i])
            count = count + 1
        i = i + 1
```

Рассмотрим интерфейс работы с программой. Сначала необходимо ввести требуемое ключевое слово, после этого происходит поиск и выводятся пронумерованные результаты. Следующим шагом будет выбор номера класса из результатов предыдущего шага, после чего программа выведет комментарий, а также предшествующие и последующие классы. Затем программа запустится заново.

```
Enter the query: вирус
1 .  onto.Компьютерный_вирус
2 .  onto.Порядок_проведения_испытаний_программных_средств_на_наличие_компьютерных_вирусов
```

3 . onto.Методы_проведения_испытаний_программных_средств_на_наличие_компьютерных_вирусов
4 . onto.ГОСТ_Р_51188—98_—_Защита_информации._Испытание_программных_средств_на_наличие_компьютерных_вирусов._Типовое_руководство.
5 . onto.Состав_работ_по_подготовке_к_проведению_испытаний_программных_средств_на_наличие_компьютерных_вирусов
6 . onto.оценку_эффективности_применяемых_антивирусных_средств;
7 . onto.поиск_вирусоподобных_фрагментов_кодов_ПС;
Enter the number: 1
onto.Компьютерный_вирус - ['программа, способная создавать свои копии (необязательно совпадающие с оригиналом) и внедрять их в файлы, системные области компьютера, компьютерных сетей, а также осуществлять иные деструктивные действия. При этом копии сохраняют способность дальнейшего распространения. Компьютерный вирус относится к вредоносным программам.']
Ancestors:
1 . onto.Определения_и_сокращения
2 . onto.ГОСТ_Р_51188—98_—_Защита_информации._Испытание_программных_средств_на_наличие_компьютерных_вирусов._Типовое_руководство.
Descendants:
Enter the query:

В результате проделанной работы была создана онтология и осуществлен ее программный анализ. Дальнейшие исследования могут быть направлены на уточнение терминов и добавление нормативных документов.

Литература:

1. Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы: Учебник. – СПб.: Изд-во «Лань», 2016. – 324 с.
2. 2. Федоров Д. Ю. Программирование на языке высокого уровня Python: учеб. пособие для прикладного бакалавриата – М.: Изд-во Юрайт, 2017. – 126 с.

1.3. Основные положения теории информационно-психологического воздействия

Федоров Д.Ю., Воробьев Т.М.

«*...важно понимать, какой силой обладают те, кто контролирует процесс выработки определений. Поэтому первый шаг в направлении установления контроля над определениями заключается в том, чтобы попытаться не уступить крайне важной терминологической территории*»

(Г. Шиллер)

Доктрина информационной безопасности (далее – Доктрина) подчеркивает актуальность проблемы информационно-психологического воздействия на личность и общество. «В соответствии с военной политикой Российской

Федерации основными направлениями обеспечения информационной безопасности в области обороны страны являются: ... нейтрализация информационно-психологического воздействия, в том числе направленного на подрыв исторических основ и патриотических традиций, связанных с защитой Отечества» [3].

Анализ ряда публикаций [2, 5, 9] показал, что оценка степени информационно-психологического воздействия на сегодняшний день носит, скорее, субъективный характер. В связи с этим возникает противоречие между необходимостью реализации Доктрины и отсутствием формализованной теории информационно-психологического воздействия.

Разрешением данного противоречия, на взгляд авторов, может стать предложенная далее теория информационно-психологического воздействия, основанием которой послужили труды проф. В. Я. Розенберга о семантических сетях знаний (далее – сетях знаний) [7, 8, 10].

Основные положения теории семантических сетей знаний профессора В.Я. Розенберга

Введем ряд определений. Прежде всего, под *мышлением* будем подразумевать отображение в мозгу человека общих существенных свойств (признаков) вещей, явлений внешнего мира (предметов мысли). Существенным признаком предмета называется тот признак, который выражает коренное, наиболее важное свойство предмета; если существенный признак отсутствует, то предмет перестает быть данным предметом.

Под *знанием* в широком смысле будем понимать субъективный образ реальности в форме понятий и представлений [4].

Понятие – это мысль, которая отображает общие и существенные признаки предметов. Понятие отражает сущность вещи, имеет характер всеобщности. Одними и теми же понятиями пользуются разные люди. Понятие возникает и существует на базе языковых терминов и фраз. Определение понятия есть такое логическое действие, в процессе которого раскрывается содержание понятия. Раскрыть содержание понятия – это значит указать его существенные признаки. Каждый предмет имеет бесконечное число признаков, и пытаться указать все признаки предмета невозможно. Определение содержит в себе лишь такие признаки, которые, являясь существенными, отграничивают понятие от других понятий.

Под *представлением* будем понимать наглядный образ предмета. Представление всегда имеет индивидуальный характер, оно может складываться из несущественных признаков [1].

> «*…обучение происходит путем добавления новых концепций и предложений в существующую систему понятий, которой обладает обучаемый. Иногда возникает вопрос о происхождении первичных (базовых) понятий. Они приобретаются детьми в возрасте от рождения до трех лет, когда они распознают законо-*

мерности в окружающем их мире и начинают распознавать языковые метки или символы для этих закономерностей. Эта способность является частью эволюционного наследия людей. После 3-х лет новые знания получают путем вопросов и выяснения отношений между старыми и новыми понятиями» [13].

(Дэвид Пол Аусубель)

Совокупность знаний отдельного человека или всего человечества образует систему знаний. В качестве элемента системы знаний проф. В. Я. Розенберг [7] предложил использовать формализм понятия и технологию построения *сети знаний*. Основная идея сети знаний заключается в построении многоуровневой сети связанных между собой понятий. Понятия связываются через определения. Вышележащие понятия можно усвоить, если усвоены понятия, лежащие на более низком уровне сети знаний. В качестве примера на рис. 1.4 представлен фрагмент сети знаний, составленный на основе понятий из области уголовного права.

Пример построения индивидуальной картины мира человека на основе сетей знаний

Человека всюду окружают информационные потоки, поэтому необходимо различать понятия «информация», как входной поток, и «знание», как переработанную информацию. Федеральный закон «Об информации, информационных технологиях и о защите информации» определяет информацию, как сведения (сообщения, данные) независимо от формы их представления. В терминах теории сетей знаний, под *знанием* будем понимать набор понятий и связей между ними. Исходя из этого определения, информация становится знанием в момент построения сети знаний.

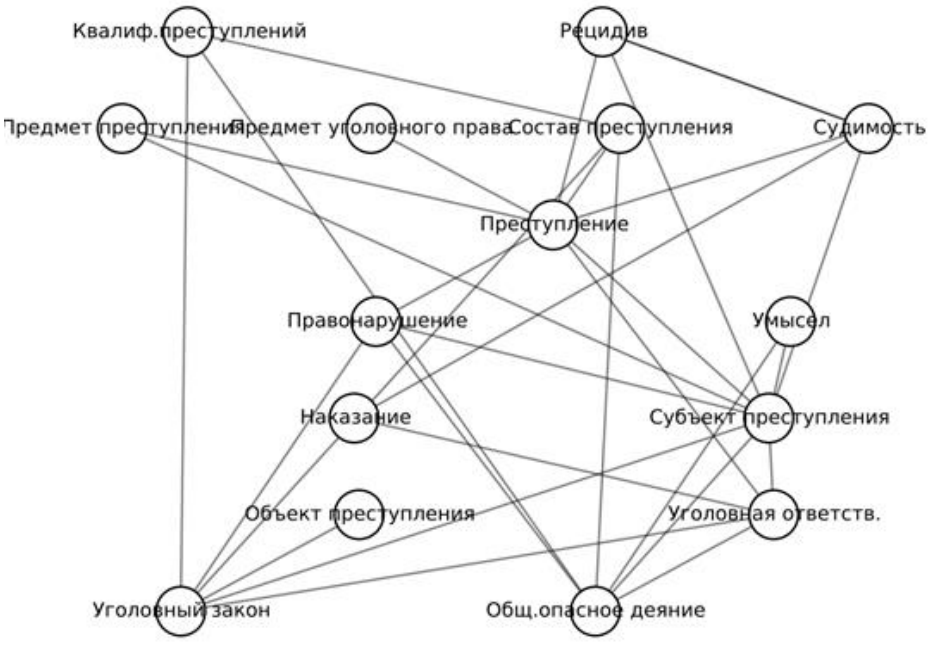


Рис. 1.4. Фрагмент сети знаний для области уголовного права

Рассмотрим область индивидуального человеческого знания $K \subset U$, где K – множество понятий и их определений из множества U – всех знаний человечества на данный момент времени [8]. Условимся, что K является упрощенной моделью индивидуальной картины мира человека.

Для автоматизации построения индивидуальной картины мира на базе семантических сетей на языке Python [12] была написана программа (автоматизированная система построения сети знаний)[2], реализующая следующий алгоритм извлечения знаний [6] (см. рис. 1.5).

Разработанная автоматизированная система построения сети знаний решает следующие задачи:

1) позволяет с клавиатуры вносить определения терминов в систему;
2) позволяет добавлять в систему новые термины и их определения;
3) по терминам и их определениям строит матрицу инцидентности однонаправленного графа (сеть знаний);
4) по матрице инцидентности строит сеть знаний.

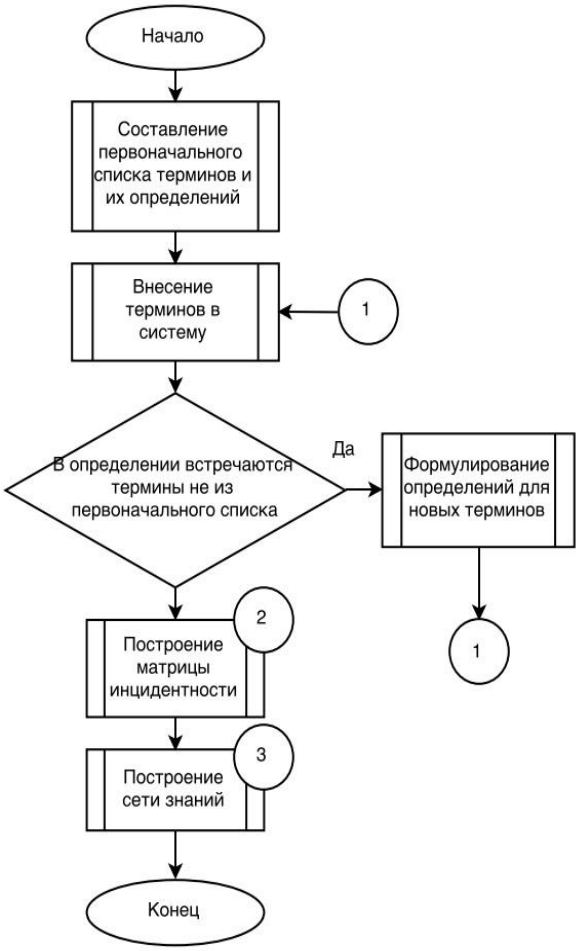


Рис. 1.5. Блок схема алгоритма извлечения знаний для построения индивидуальной картины мира человека (сети знаний)

[2] Программа реализована студентом направления «Прикладная математика и информатика» СПбГЭУ Т.М. Воробьевым.

Для реализации обозначенных задач используются следующие библиотеки на языке Python: string для работы со строками, copy позволяет копировать объекты, numpy для работы с большими массивами, matplotlib позволяет визуализировать ориентированный граф (сеть знаний), networkx для создания и обработки графов, pymorphy2 и nltk для обработки текстов, pymongo для работы с базой данных.

В основе работы автоматизированной системы построения сети знаний лежит клиент-серверная архитектура. Добавление терминов в систему производится через веб-интерфейс.

Перейдем к рассмотрению реализации основных элементов системы.

```
# составляем список из терминов, для которых потребуется
# сформулировать определения:
s1 = ['дерево свойств', 'интегративные свойства', 'качество', 'объект', 'поведение', 'подсистема', 'развитие', 'свойство', 'связь', 'система', 'ситуация', 'состояние', 'среда', 'структура', 'управление', 'функционирование', 'характеристика', 'целое', 'цель', 'элемент']
```

Исходный код обращения к базе данных для считывания терминов и их определений, введенных пользователем:

```
client = MongoClient('localhost', 27017)
db = client['main']
collection = db['laba']
s1 = []
s2 = []
nul = []
# содержит список терминов:
s1=list(collection.find_one()['dicts'][1].keys())
# содержит список определений:
s2=list(collection.find_one()['dicts'][1].values())
```

Удалим из всех терминов и их определений знаки пунктуации и не несущие нагрузки слова при помощи методов stopwords.words(«russian») и string.punctuation. Далее преобразуем все определения в список слов, входящих в них, и переведем все слова в нормальную форму. Это позволяет сделать функция tokenize:

```
# на вход функции подается строка, содержащая термин
# или его определение
def tokenize(s):
    # разбивает строку, написанную на русском языке,
    # на список из слов и знаков пунктуации, входящих в эту строку:
    tokens = nltk.word_tokenize(s,language='russian')
    prepositions = stopwords.words("russian")
    # создает новый список из слов, являющихся
    # нормальными формами слов, входящих в строку,
    # не входящих в список prepositions и не
```

```
    # являющимися пунктуационными знаками:
    tokens = [morph.parse(i)[0].normal_form for i in tokens if ( i not in
    prepositions and i not in string.punctuation)]
    return tokens
```

Вызов функции tokenize производится следующим образом:

```
for i in range(len(s1)):
        s1[i]=tokenize(s1[i])
        s1[i]=' '.join(s1[i])
        # превращает термин в строку, состоящую из слов
        # в нормальной форме:
        s2[i]=tokenize(s2[i])
```

Матрицей инцидентности однонаправленного графа называется матрица, в которой на позиции [i][j] стоит 1, если элемент i связан с элементом j или 0 – в противном случае. В данном случае на позиции [i][j] стоит 1, если определение термина j содержит термин i или 0 – в противном случае. Реализация построения матрицы (изначально матрица нулевая) имеет следующий вид:

```
nul = [0 for i in range(len(s1))]
matrix = []
labels = {}
for i in range(len(s1)):
    nule = copy.copy(nul)
    # создание нулевой матрицы размера len(s1) на len(s1):
    matrix.append(nule)
    labels[i] = s1[i]
    s1[i] = tokenize(s1[i])
    s1[i] = ' '.join(s1[i])
    s2[i] = tokenize(s2[i])
for i in range(len(s1)):
    for j in range(len(s2)):
        if (i != j):
            if len(s1[i].split()) == 1:
                for k in range(len(s2[j])):
                    if s1[i] in s2[j][k]:
                        matrix[i][j] = 1
                    elif stemmer.stem(s1[i]) in s2[j][k]:
                        c = input('Относиться ли слово '+s2[j][k]+' к термину '+
labels[i]+'\nДа-1\nНет-2\n')
                        if c == '1':
                            matrix[i][j] = 1
            else:
                for k in range(len(s2[j])-len(s1[i].split())+1):
                    s3 = ''
                    for n in range(len(s1[i].split())):
                        s3 = s3 + ' ' + s2[j][k+n]
```

```
if s1[i] in s3:
    matrix[i][j] = 1
    break
```

Заметим, что отдельно рассматривается случай, когда термин составной. При данном решение возникают проблемы со сложными словами, например, со словом «целенаправленный». Также проблемы возникают со словами, которые образованы от терминов, например, «целостный» или «связанный», так как метод normal_form возвращает слово в нормальной форме, то есть слово «связанное» преобразуется в слово «связанный» или «связать», но не в слово «связь». Для решения этой проблемы пришлось воспользоваться методом stemmer.stem(s), который возвращает основу слова, например, «целый» преобразуется в слово «цел». Но возникает новая проблема, так как термины, например, «целый» и «цель» имеют одинаковые основы. Для решения этой проблемы пришлось запрашивать у пользователя, к какому термину относится, например, слово «целостный» – «цель» или «целый».

Далее представлен пример обработки терминов пользователем:

```
Относится ли слово связанный к термину связь
Да
Нет
Пользователь - Да
Относится ли слово целостный к термину целое
Да
Нет
Пользователь - Да
Относится ли слово целенаправленный к термину целое
Да
Нет
Пользователь - Нет
Относится ли слово посредством к термину среда
Да
Нет
Пользователь - Нет
Относится ли слово целостный к термину цель
Да
Нет
Пользователь - Нет
Относится ли слово целенаправленный к термину цель
Да
Нет
Пользователь – Да
```

В начале процесса построения сети знаний создается ориентированный граф G и в него добавляется количество вершин равное длине матрицы. Хранить информацию о каждой вершине будем в списке graf, где:

graf[i][0][0] – уровень i-ой вершины (изначально равен ‘0’),
graf[i][1] – список вершин, связанных с вершиной i,
graf[i][2] – список вершин, с которыми связана вершина i.

```
G=nx.DiGraph()
G.add_nodes_from([i for i in range(len(matrix))])
# словарь, который будет хранить координаты в виде pos[i]=(x, y)
pos={}
# переводит список matrix в массив:
matrix=numpy.array(matrix)
graf=[]
for i in range(len(matrix)):
    graf.append([['0'],[],[]])
# максимальный уровень графа
maximum=0
```

Заметим, что в начале программы был создан словарь labels, который хранит не отформатированные термины – названия вершин графа:

```
labels = {}
for i in range(len(s1)):
    labels[i] = s1[i]
```

Необходимо найти все корни, то есть элементы, с которыми никакие другие элементы не связаны. Из определения матрицы инцидентности однонаправленного графа следует, что элемент j является корнем, если j-ый столбец матрицы инцидентности нулевой. Найдем все такие элементы и присвоим им нулевой уровень, а список входящих вершин изменим на [‘Нет’]:

```
j=0
for i in matrix.transpose().tolist():
    if 1 not in i:
    graf[j][0][0]=0
    graf[j][1].append('Нет')
    pos[j]=(j,0)
    j=j+1
```

Далее, опираясь на правила построения сети знаний, формируем ориентированный граф:

а) каждая вершина должна находиться на определенном уровне;

б) если вершина i связана с вершиной j, то вершина j должна, по крайней мере, находиться на один уровень выше уровня вершины i;

в) если вершина i не связана ни с какой из вершин j и не находится на последнем уровне, то ее необходимо переместить на последний уровень.

Граф строится при помощи обхода в ширину. Счетчик k обозначает, какой уровень рассматривается. Если элемент j лежит на уровне k, то рассматриваем все i-ые вершины, с которыми связана вершина j. Затем в граф

G добавляем ориентированное ребро – G.add_edge(j, i), где j – начало, а i – конец и обновляем списки выходящих вершин для вершины j и список входящих вершин для вершины i. Если уровень вершины i больше уровня вершины j, то ничего не делаем. В противном случае, так как граф должен соответствовать пункту б, то применяем к i вершине рекурсивную функцию up, которая поднимает все поддерево с корнем в вершине i на величину равную разности уровней вершин j и i плюс один:

```
def up(i,q):
    global graf,maximum,pos
    graf[i][0][0] = q+1
    pos[i] = (i,graf[i][0][0])
    if graf[i][2] != []:
        for j in graf[i][2]:
            up(j,graf[i][0][0])
    elif graf[i][0][0]>maximum:
            maximum = graf[i][0][0]
j=0
for i in matrix.transpose().tolist():
    if 1 not in i:
        graf[j][0][0]=0
        graf[j][1].append('Нет')
        pos[j]=(j,0)
    j=j+1
k=0
while k<=maximum:
    for j in range(len(matrix)):
        if graf[j][0][0]==k:
            for i in range(len(matrix)):
                if matrix[j][i]==1:
                    G.add_edge(j,i)
                    graf[j][2].append(i)
                    graf[i][1].append(j)
                    if int(graf[i][0][0])<=graf[j][0][0]:
                        up(i,graf[j][0][0])
    k=k+1
for i in range(len(graf)):
    if graf[i][2]==[]:
        graf[i][0][0]=maximum
        pos[i]=(i,maximum)
```

Также рассматриваем уровень верхних элементов поддерева в корне i, после их поднятия, и сравниваем его со значением maximum, которое отвечает за значение максимального уровня графа. Последний цикл, удовлетворяет требованию в. Он поднимает все вершины, не имеющие детей, на максимальный уровень.

Улучшение визуализации ориентированного графа:

```
for i in range(maximum+1):
    l=[]
    for j in range(len(graf)):
        if pos[j][1]==i:
            l.append(pos[j])
    j=0
    while j <len(l)-1:
        if l[j+1][0]-l[j][0]<3:
            pos[l[j+1][0]]=(pos[l[j][0]][0]+3,i)
            l[j+1]=(l[j][0]+3,i)
            l.sort()
        else:
            j=j+1
```

Визуализации ориентированного графа:

```
nx.draw_networkx(G,pos,with_labels=False,node_size=2000,node_color='w',
node_shape='s')
nx.draw_networkx_labels(G,pos,labels,font_size=8)
plt.axis('off')
plt.show()
```

Результат выполнения программы представлен на рис. 1.6, 1.7 и 1.8.

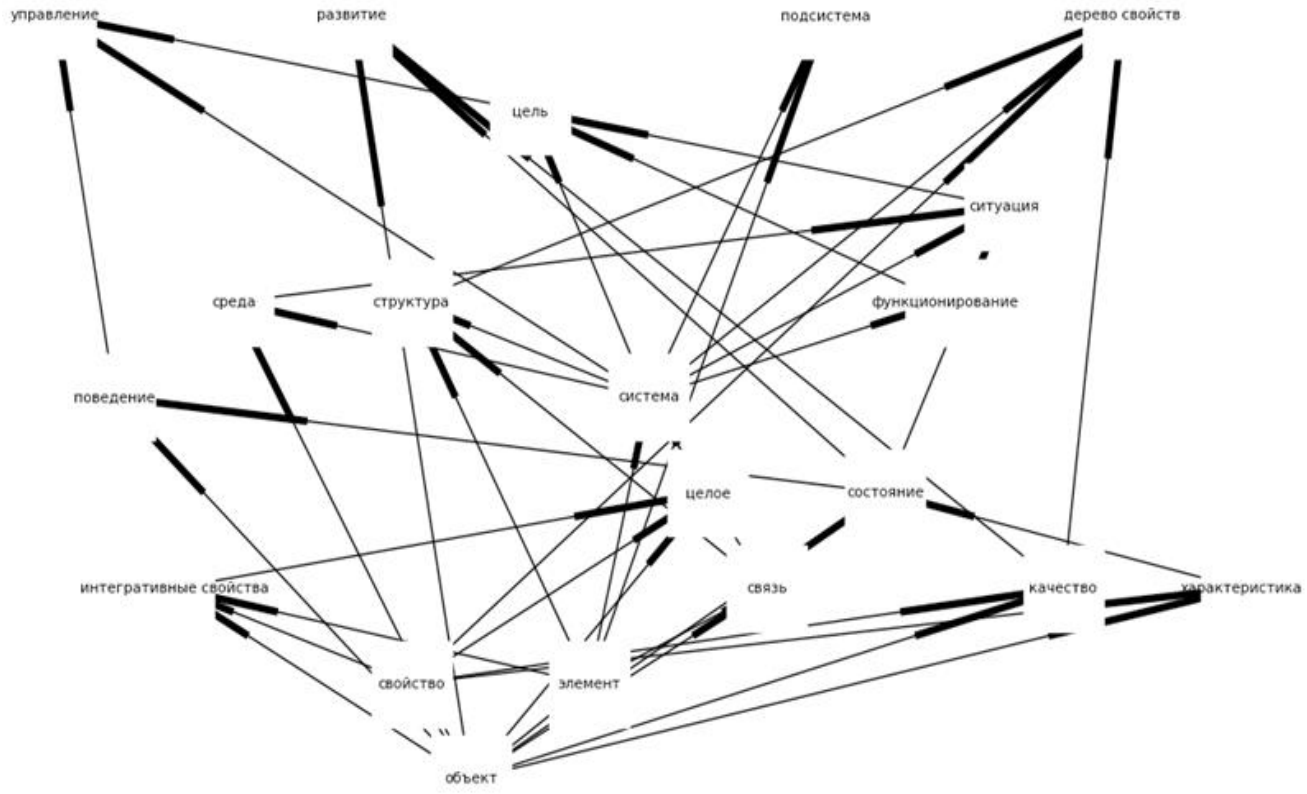


Рис. 1.6. Пример построения сети знаний для терминов
из предметной области «Теория систем»

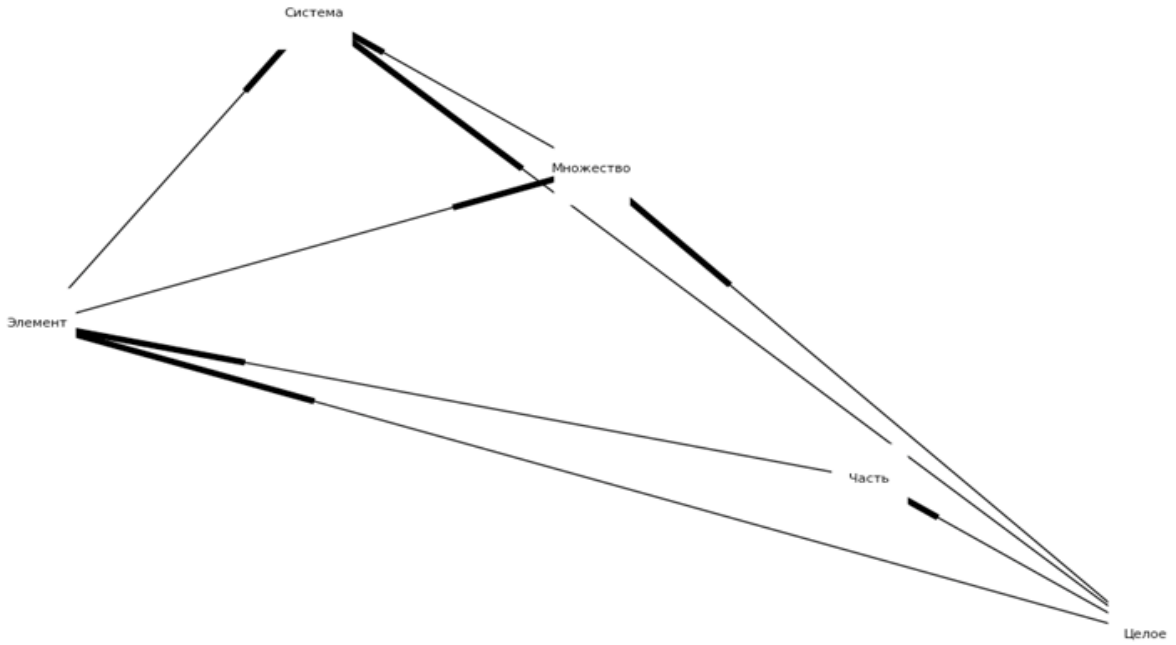


Рис. 1.7. Пример построения сети знаний

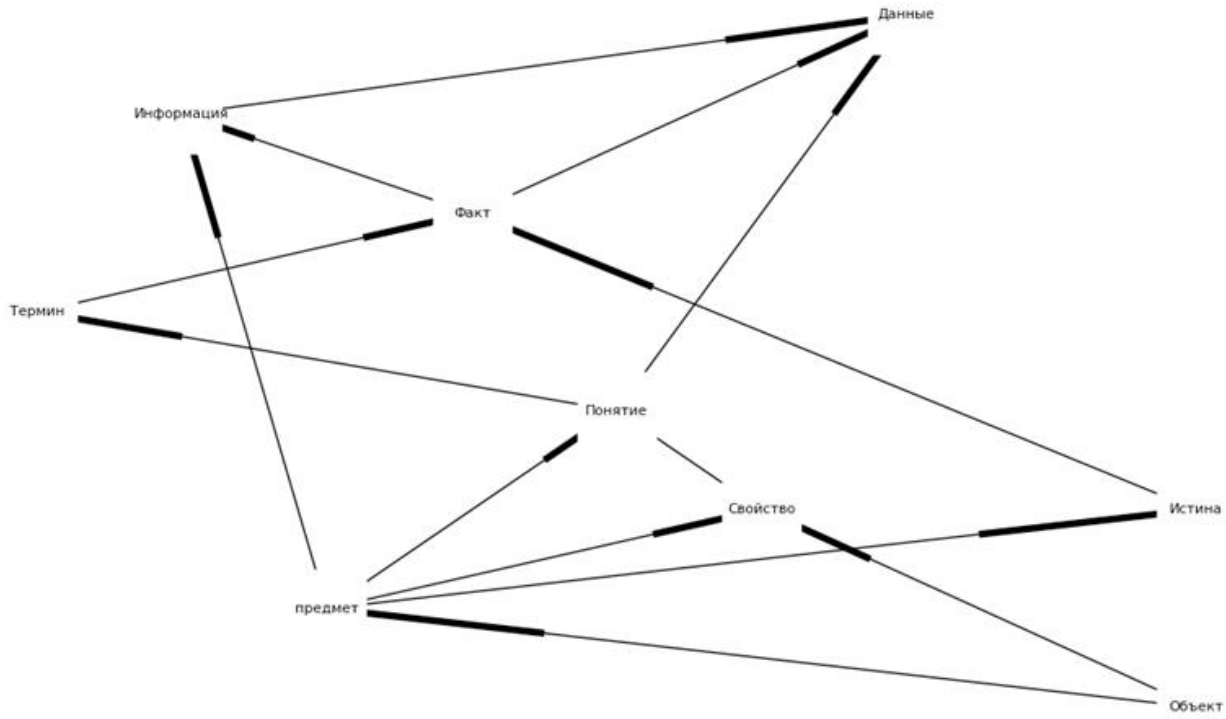


Рис. 1.8. Пример построения сети знаний

Основные положения теории информационно-психологического воздействия

Субъект (человек, группа людей) по средствам информации *T* (в частности, через текст) производит воздействие на объект (человека, группу людей), который данную информацию воспринимает посредствам органов

чувств. Информация формируется таким образом, чтобы максимально воздействовать (деформировать, исказить) на модель знаний *K* (индивидуальную картину мира), содержащуюся в голове объекта воздействия. Исходя из этого, можно сформулировать понятие *количества энтропии информации* (текста), как величину, определяющую степень деформации модели знаний объекта (*z-энтропия*).

Рассмотрим способы осуществления информационно-психологического воздействия на примере сетей знаний, в основу которых положена классификация Леонтьева А. А. [11].

1. «Ввести в поле значений (в контексте сети знаний, картину мира *K*) реципиента новые значения, сообщить ему такие новые знания о действительности, на основе которых он изменит свое поведение или, по крайней мере, свое отношение к этой действительности». Примером такого воздействия может быть новость о взрыве во время проведения Бостонского марафона 15 апреля 2013 года. Это событие актуализировало понятие, которое в картине мира реципиента ассоциировалось со взрывом. Например, в зависимости от индивидуальных представлений ассоциации могли быть следующими: «теракт для устрашения», «злая шутка», «заговор спецслужб», «месть воинов джихада» и др. В результате, актуализированное понятие может стать мишенью для дальнейшей манипуляции.

2. «…изменить поле значений реципиента, не вводя в него новых элементов, т.е. изменить понимание реципиентом событий и их взаимосвязи. Это тоже информирование, но на другом уровне, когда событие уже известно, но благодаря воздействию оно интерпретируется реципиентом по-другому». Примером такого воздействия может быть сообщение об организаторах Бостонского теракта. Со слов председателя Комитета Палаты представителей по национальной безопасности: «Поездка Тамерлана Царнаева в северо-кавказский регион, радикальные видео о провозглашении халифата, которые он разместил в Интернете по возвращении, взрывные устройства, которые были использованы им и его младшим братом, – все это говорит о том, что этот теракт был инспирирован «Аль-Кайдой»». Реципиенту известен факт совершения взрыва (см. п.1), ранее было известно о существовании «Аль-Кайды». Через информационное сообщение устанавливается связь между двумя понятиями: актуализированным после новости о событии и ранее известным – «Аль-Кайда». Например, в зависимости от актуализированного понятия в п.1, «Аль-Кайда» в индивидуальной картине мира реципиента может связываться с «террористами, действующими для устрашения», «злой шуткой», «заговором спецслужб», «воинами джихада» и др.

3. В картину мира реципиента вводится новый элемент, который отсутствовал там ранее. Например, сообщение о братьях Царнаевых, подозреваемых в организации взрывов на Бостонском марафоне. Впоследствии с новым элементом устанавливается связь, аналогичная п.2. Таким образом, картина мира реципиента достраивается новыми элементами.

4. В картине мира реципиента старое понятие подменяется новым, т.е. известное понятие обогащается информацией, ранее о которой не было известно реципиенту. Например, Плутон, когда-то был девятой планетой, а сегодня является представителем нового семейства планет-карликов. Изменилось содержание понятия «планета».

5. В картине мира реципиента старое понятие вытесняется новым, не схожим по смыслу со старым. Например, казнокрадство именуется «нецелевым расходованием бюджетных средств». Замещение понятия приводит к перестроению ассоциативного ряда реципиента. Новое понятие не несет негативной окраски, присущей старому понятию.

На основании алгоритма построения сети знаний можно ввести определение ранжирования понятия. Под *ранжированием* будем понимать определение порядка понятий согласно их авторитету. Авторитет понятия формируется по количеству вхождений в определения других понятий из области знаний K. Наиболее авторитетные понятия составляют множество $A \in K$ и будут являться системообразующими для области знаний K [11].

Любое воздействие на множество K переводит его из состояния S_1 в состояние S_n, где S_n – набор актуализированных понятий на данный момент времени. Приведенная классификация видов информационного воздействия является неполной, но позволяет построить модель формирования индивидуальной картины мира K под воздействием поступающей информации T. Главной мишенью информационного воздействия является множество системообразующих понятий A, изменение которых приведет к перестроению множества K, т.к. системообразующие понятия включаются в определения большого числа других понятий.

Следует отметить, что рассмотренные положения теории информационно-психологического воздействия находятся на начальном этапе формирования[3].

Литература:

1. Виноградов С.Н., Кузьмин А.Ф. Логика. Учебник для средней школы. – М.: Государственное учебно-педагогическое изд-во министерства просвещения РСФСР, 1954. – 176 с.

[3] Список публикаций по теме исследования: http://dfedorov.spb.ru/science.html

2. Володенков С.В. Интернет-коммуникации в глобальном пространстве современного политического управления. – М.: Изд-во Московского ун-та, 2015. – 320 с.
3. Доктрина информационной безопасности РФ № Пр-646 от 5 декабря 2016 г.
4. Знание. Статья в Википедии [Электронный ресурс]. URL: http://ru.wikipedia.org/wiki/Знание (дата обращения 17.10.2017).
5. Макаренко С.И. Информационное противоборство и радиоэлектронная борьба в сетецентрических войнах начала XXI века. Монография. – СПб.: Наукоемкие технологии, 2017. – 546 с.
6. Подружкина Т.А., Федоров Д.Ю. Алгоритмы планирования процесса обучения на основе семантических сетей знаний // Научно-аналитический журнал «Вестник Санкт-петербургского университета ГПС МЧС России». – 2017. – № 1. – С. 107-116.
7. Розенберг В.Я. Аксиомы математической теории исчисления знаний [Электронный ресурс]. URL: http://passat.spb.ru/webpassat1/wp-content/uploads/2015/08/TEORIYA-ISCHISLENIYA-ZNANIY.pdf (дата обращения 11.11.2017).
8. Розенберг В.Я. Система обучения на базе семантических сетей. Теория и практика // Фундаментальные и прикладные исследования в современном мире: материалы Междунар. науч.-практ. конф. – СПб.: Информ. изд. учеб.-науч. центр «Стратегия будущего», 2013. – С. 184-191.
9. Соловей В.Д. Абсолютное оружие. Основы психологической войны и медиаманипулирования. М.: Издательство «Э», 2015. – 320 с.
10. Федоров Д.Ю. Кибернетический подход к управлению процессом обучения на основе семантических сетей знаний. – СПб.: Изд-во Политех. ун-та, 2016. – 40 с.
11. Федоров Д.Ю. Применение структуризации знаний для обеспечения информационной безопасности личности // Национальная безопасность и стратегическое планирование. – 2013. – № 2. – С.23-27.
12. Федоров Д.Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. – М.: Изд-во Юрайт, 2017. – 126 с.
13. Novak Joseph D. Learning, Creating, and Using Knowledge: Concept maps as facilitative tools in schools and corporations – London; New York: Routledge, 2010. – 317 p.

1.4. Использование теории графов для проектирования стратегических задач виртуального предприятия

Соколовская С. А.

Интернет и коммуникационные технологии активно используются во все областях, создавая мировую сетевую экономику с глобальными рынками, в которых постоянно происходят изменения, а инновации и уникальные продукты и услуги становятся особенно важными, вытесняя массовые. Поэтому, одной из главных причин возникновения концепции виртуальных предприятий можно считать развитие и широкое распространение современных Интернет-технологий, дающих возможность для обмена информацией и совместной деятельности различных автономных, географически распределенных предприятий.

Виртуальное предприятие (ВП) можно рассматривать как динамическую открытую бизнес-систему, организованную юридически независимыми предприятиями с помощью общего информационного пространства, задачей которого является совместное использование технологических средства производства для реализации всех этапов работ по выполнению проекта.

Для проектирования бизнес-процессов стратегических задач управления виртуальным предприятием можно использовать теорию графов, при этом предполагаемая структура ВП должна быть прозрачной для всех участников, а на концептуальном уровне ее можно представить, как факт наличия отношения или набора отношений между двумя или более экономическими агентами (рис. 1.9). Причем, отношения между агентами виртуального предприятия можно рассматривать с помощью ребер графа, вершинами которого будут сами агенты предприятия.

Если предположить, что еще не установлено, как именно должен вести себя экономический агент и в каком отношении он будет состоять то, можно констатировать только факт существования отношения, когда агент 1 имеет связь с агентом 2 и не имеет ее с агентом 3 (см. рис. 1.9).

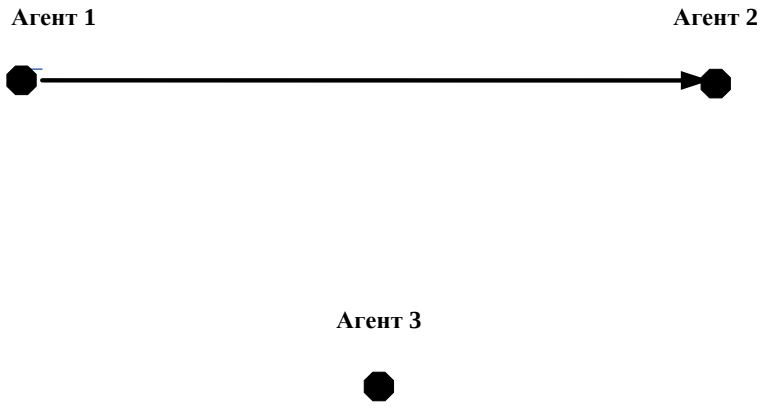


Рис. 1.9. Наличие отношений между экономическими агентами в рамках ВП

Для первичного установления наличия или отсутствия эмпирически улавливаемых отношений можно использовать инструмент отслеживания связей. Например, при установлении/отсутствии связей между агентами ВП (рис. 1.10), можно увидеть, что агент А имеет 5 таких отношений, В имеет 4, а С – только 2 и агент М не имеет отношений ни с кем из наблюдаемой совокупности.

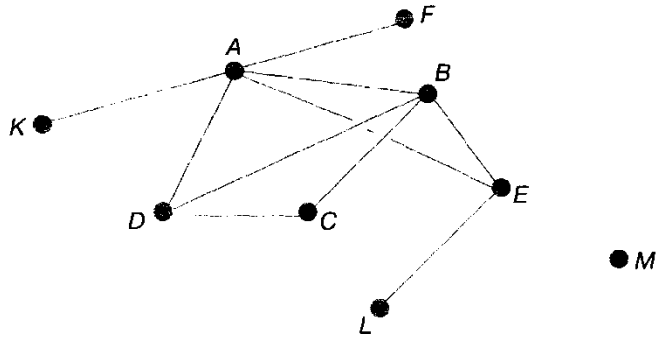


Рис. 1.10. Сетевая структура ВП

Детальное изучение связей позволяет фиксировать частоту взаимодействий (трансакций) между агентами ВП. Можно сделать вывод, что лидером по числу трансакций является агент В, а самая устойчивая связь – между агентами В и С (рис. 1.11).

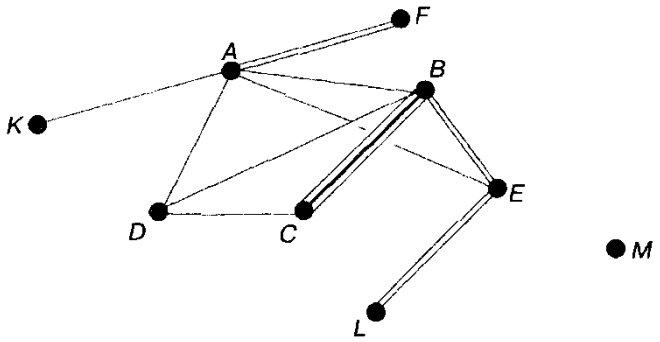


Рис. 1.11. Частота трансакций в рамках ВП

Если предположить, что взаимоотношения для экономических агентов не бесплатны, то не бесплатна и информация о потенциальных агентах – партнерах. Следовательно, конструкции связей между ними, единожды сложившись, будут иметь больше шансов на воспроизведение, чем на замену совершенно новым набором связей. При этом чем выше затраты, обеспечивающие взаимодействия, тем более стабильной является структура связей в рамках ВП.

При нулевых затратах происходит обновление около 50% связей (худшей половины), и ведется активный поиск новых, лучших вариантов (см. рис. 1.12а). Если затраты существуют, но они относительно низки, идет процесс обновления только явно неудовлетворительных связей (рис. 1.12б). Если же затраты высоки, отказ от плохих связей ведет не к формированию новых, а к учащению хороших (рис. 1.12с) [1].

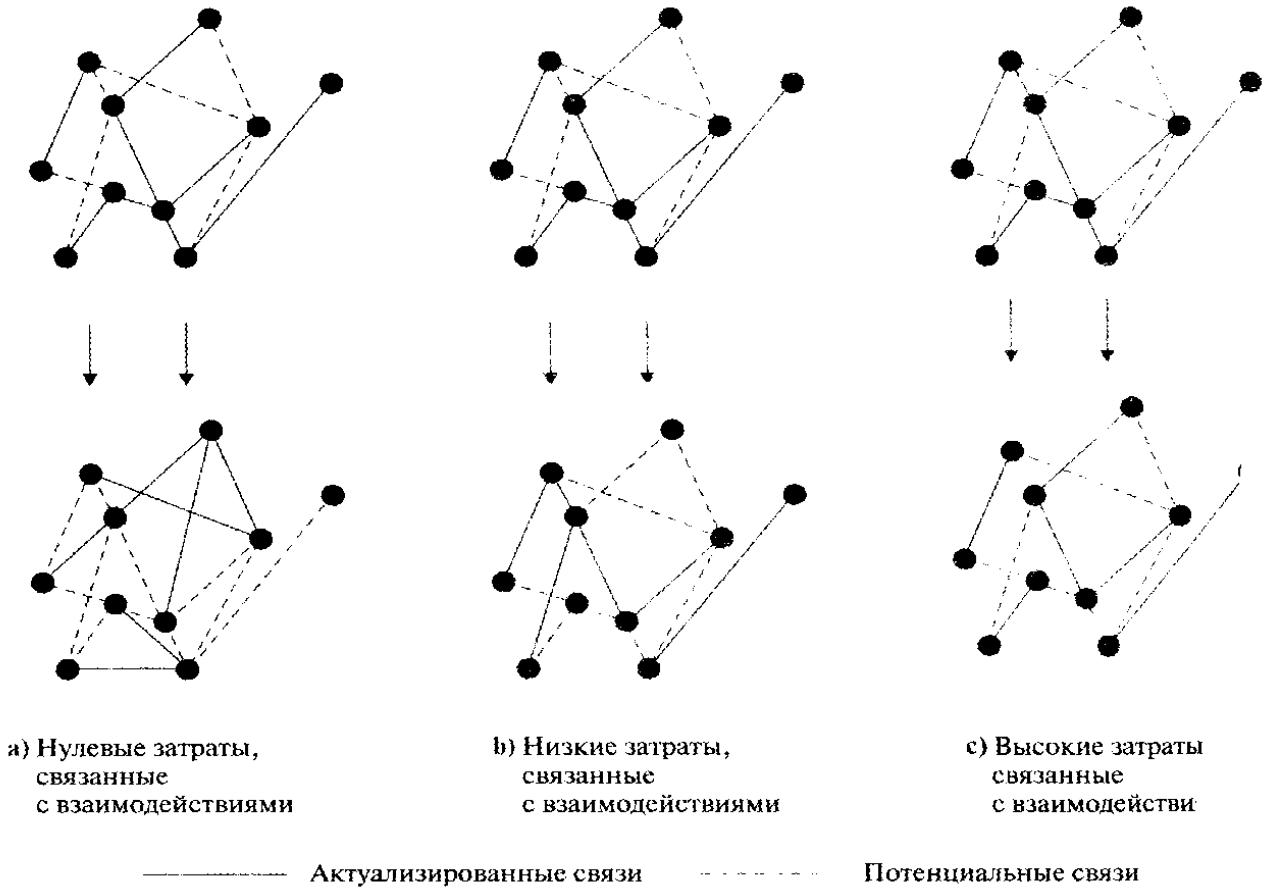


Рис. 1.12. Затраты, связанные с взаимодействиями, актуализация связей в рамках ВП

Деятельность ВП направлена на создание определенной услуги/продукта, что и влияет на выбор экономических агентов, способствуя формированию устойчивых связей между ними. Причем эти связи на время жизненного цикла ВП приводят к возникновению устойчивых структур отношений для достижения поставленных целей. Формой представления ВП как раз и могут служить такие структуры, или сети.

Основными объектами ВП выступают агенты, преследующие определенные стратегические цели. Поэтому используя теорию графов можно построить дерева целей, которое представляет собой графическую схему декомпозиции общих целей на подцели.

Все цели ВП можно рассматривать в разрезе четырех основных проекций задач ВП:

- финансовый аспект;
- аспект агента;
- внутрифирменный аспект;
- аспект инноваций и обучения.

Для эффективного управления ВП необходимо ранжировать цели по степени их важности и, следовательно, по приоритетности их достижения. Для решения задач управления проводится анализ причинно-следственных связей на предмет оценки важности каждой цели по ее взаимодействию и влиянию на другие цели.

Таким образом, отношения экономических агентов – людей, организаций, стран, складывающиеся в рамках ВП, имеют свою структуру.

Как правило, недостаточно стабильная структура ВП, созданного на период функционирования, предсказуема на короткий момент времени, то есть может являться предметом исследования. Причем о структуре ВП можно говорить как о сети, связывающей между собой агентов (участников) отношений.

Впервые принципы теории сетей сформулировал Я. Морено, в 1934 г. Сеть – это совокупность связей между группой экономических агентов, которые находятся друг с другом в тех или иных отношениях. Сети могут участвовать в любых взаимодействиях. Центральным элементом сети будет структура отношений.

Для каждого конкретного набора участников в рамках ВП в зависимости от цели управления анализируются разные отношения, поэтому и сети будут различными.

При изучении сетей как феномена, исследователи уделяют основное внимание содержанию и структуре связей между экономическими агентами. Используя сети как инструментарий, исследователи оценивают силу и частоту связей, возможности участников отношений и пр. (табл. 1.5).

Отношения между агентами в рамках ВП могут быть как направленными, так и ненаправленными. Для анализа можно применять направленные (ориентированные) графы или ненаправленные (неориентированные).

Направление взаимодействия в виде ориентированного графа может быть как от низшего к высшему уровню управления, а так же допустимо влияние на одном уровне.

С помощью ориентированного графа можно описать взаимодействия между агентами ВП, их цели. В качестве направления связей можно рассматривать пути достижения целей управления ВП.

С точки зрения теоретико-множественного описания, структура причинно-следственных взаимодействий целей управления ВП в виде графа

связей представляет собой ориентированный граф G, нагруженный весами дуг E, т.е. $G=(X,L)$, где X – множество вершин графа, представляющих цели виртуального предприятия, L - ребра графа G, определяющие взаимодействие целей (рис. 1.13).

Таблица 1.5

Анализ сетей

Действие	Пояснение
объяснить	неформальные отношения между экономическими агентами влияют на результаты деятельности ВП
оценить	потенциальную эффективность тех или иных институциональных реформ и нововведений
рассчитать	оптимальную структуру ВП
учесть	связи при институциональном проектировании
определить	какие агенты являются центральными для отношений, а какие – периферийными

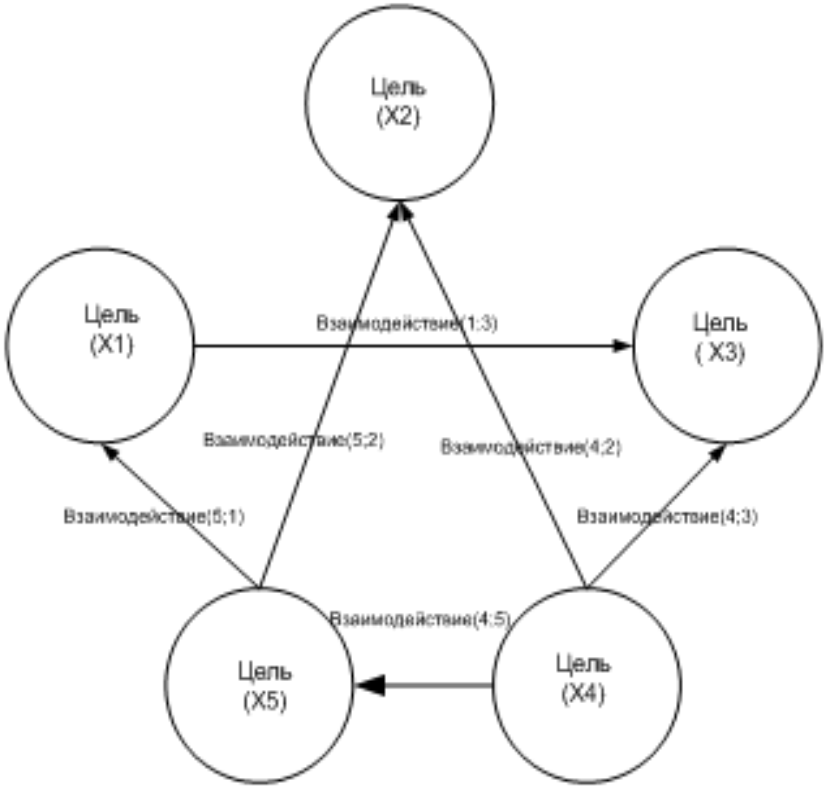


Рис. 1.13. Структура причинно-следственных взаимодействий целей управления ВП в виде ориентированного графа связей

Граф изображается точками на плоскости и линиями $l_k=(n_{i,}n_j)$, соединяющие эти точки. На рис. 1.13 показан граф, имеющий 5 вершин и 6 ребер.

Пусть рассматривается множество упорядоченных пар $l_k = (n_{i,} n_j)$из множества точек $X = (x_1,....., x_g)$, на каждом ребре из множества $L = \{l_1,....., l_z\}$ задается направление, то граф, $G = (X, L)$ называется ориентированным. Если же на каждом ребре из множества $L = \{l_1,....., l_z\}$ направление не задается, то граф $G = (X, L)$ называется неориентированным графом, или просто графом.

Для каждой вершины-цели в ориентированном графе есть входящие и исходящие ребра. Поэтому можно говорить о степени вершины-цели, которая является значимой характеристикой экономических взаимодействий достижения целей в рамках ВП. Чем меньше степень вершины-цели, тем меньше влияние других целей на конкретную цель в описываемым графе.

Для характеристики отношений, связанных с конкретной вершиной-целью, используют два показателя средней степени вершин-целей $\overline{d}_{in}$ и $\overline{d}_{out}$:

- степень захода d_{in}, которая равна числу ребер, входящих в вершину-цель;
- степень исхода d_{out}, которая равна числу ребер, исходящих из вершины-цели.

Показатели средней степени вершин-целей можно рассчитать по формуле 1:

$$\overline{d}_{in} = \frac{\sum_{i=1}^{g} d_{in}(n_i)}{g}$$

$$\overline{d}_{out} = \frac{\sum_{i=1}^{g} d_{out}(n_i)}{g}$$

где

$\overline{d}_{in}$ – средняя степень вершин-целей для совокупности всех входящих графа $G = (X, L)$;

$\overline{d}_{out}$ – средняя степень вершин-целей для совокупности всех исходящих графа $G = (X, L)$;

g – общее число вершин-целей графа;

$d_{in}(n_i)$ – степень вершины-цели n_i, которая равна числу ребер, входящих в вершину-цель;

$d_{out}(n_i)$ – степень вершины-цели n_i, которая равна числу ребер, исходящих из вершины-цели.

Их дисперсии $\sigma^2_{d_{in}}$ и σ^2_{out} равны:

$$\sigma^2_{d_{in}} = \frac{\sum_{i=1}^{g} \left(d_{in}(n_i) - \overline{d}_{in}\right)^2}{g}$$

$$\sigma_{out}^2 = \frac{\sum_{i=1}^{g}\left(d_{out}(n_i) - \bar{d}_{out}\right)^2}{g}$$

Вариация как мера разброса вершин-целей на единицу степени вершин-целей равны:

$$V_{\sigma_{d_{in}}} = \frac{\sigma_{d_{in}}}{\bar{d}_{in}} \cdot 100\%$$

$$V_{\sigma_{d_{out}}} = \frac{\sigma_{d_{out}}}{\bar{d}_{out}} \cdot 100\%$$

А поскольку общее число входящих ребер равно общему числу исходящих, то:

$$\bar{d}_{in} = \bar{d}_{out} = \frac{L}{g}$$

Плотность ориентированного графа характеризуется коэффициентом плотности Δ – отношением числа ребер в анализируемом графе к числу ребер в полном графе:

$$\Delta = \frac{L}{g(g-1)}$$

Коэффициент плотности Δ варьируется в промежутке $0 \le \Delta \le 1$. Полному графу соответствует единичная плотность, а графу, в котором все вершины-цели изолированы – нулевая,

Соответственно, плотность графа будет отражать активность связей, то есть долю реализованных целей ВП.

Если для каждой вершины-цели графа $G = (X, L)$, имеющей хотя бы одну входящую дугу из множества E построить множество влияющих вершин-целей X, то получится гиперграф $H = (X, E)$, где X – множество вершин-целей гиперграфа, E – множество его ребер. При этом E представляет собой все подмножества влияющих вершин-целей X, включая вершину-цель, на которую они влияют.

Гиперграфу $H = (X, E)$ можно взаимно однозначно поставить в соответствие:

$$B(H) = \langle F, X, E \rangle$$

где

X – область отправления $B(H)$, совпадающая с множеством вершин-целей гиперграфа H;

E – область прибытия $B(H)$, совпадающая с множеством ребер гиперграфа H;

F – множество пар (x_j, e_i): $j \in J, i \in I$, образующих график соответствия $B(H)$, причем $(x_j, e_i) \in F$, если $x_j \in e_i$ в гиперграфе H. Здесь J и I размерности множеств вершин-целей и ребер гиперграфа соответственно.

Важным фактором формирования реальных связей является наличие доверия между агентами, которое приводит к возникновению потенциальных связей между ними.

Примеры реляционных связей, возникающих в рамках ВП:

- трансфер ресурсов;
- ассоциация;
- аффилирование;
- общность целей;
- обмен информацией, знаниями, взаимное обучение;
- формальные обязательства;
- доверие к партнерам.

В соответствии с рекомендациями методики *BSC* целесообразно использовать пять типов связей, оценивающих степень влияния одной связи или цели на другую:

-]0;0,2] – имеет очень слабое влияние;
-]0,2;0,4] – имеет слабое влияние*;*
-]0,4;0,6] – имеет нормальное влияние;
-]0,6;0,8] – имеет сильное влияние;
-]0,8;1] – имеет очень сильное влияние.

Лингвистической переменной называется пятерка $\{N, T, X, G, M\}$, где:

N –имя переменной;

T – терм-множество N, то есть совокупность ее лингвистических значений;

X – универсальное множество с базовой переменной x;

G – синтаксическое правило, которое может быть задано в форме бесконтекстной грамматики, порождающей термы множества T;

M – семантическое правило, которое каждому лингвистическому значению t ставит в соответствие его смысл $M(t)$, причем $M(t)$ обозначает нечеткое подмножество множества X.

Значениями лингвистической переменной являются нечеткие множества, символами которых являются слова и предложения в естественном или формальном языке, служащие, как правило, некоторой элементарной характеристикой явления.

Можно считать, что для нашего случая имеет место лингвистическая переменная, принимающая пять возможных значений. Равномерно отображая значения этой переменной в интервале [0, 1], получены числовые оценки степени влияния (нагрузки на дуги графа G): L={0,2; 0,4; 0,6; 0,8; 1}.

Учитывая, что дуги графа $G=(X,L)$ нагружены, аналитически график F соответствия $B(H)$ может быть определен как:

$$F=\{\mu(x_j,e_i),(x_j,e_i)\}: j\in J, i\in I$$

где

$\mu(x_j,e_i)$ – расплывчатый инцидентор, определяющий расплывчатую инцидентность вершины x_j ребру e_i.

Поскольку степень инцидентности в гиперграфе H стала расплывчатой, благодаря введению расплывчатого инцидентора $\mu(x_j,e_i)$, гиперграф H преобразовался в гиперграфоид $H^0(F,X,E)$, который и служит основанием для оценки степени достижения каждой цели в рамках заданной диаграммы причинно-следственных связей.

Анализ всех возможных случаев подмножества вершин-целей гиперграфоида $H^0(F,X,E)$, образованных его ребрами $e_i\in E$, показывает:

$$s_1=E(x_1)=\{\mu_q^1(x_1.e_q^1),x_q\},q\in J;$$
$$s_2=E(x_2)=\{\mu_q^2(x_2.e_q^2),x_q\},q\in J;$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$s_j=E(x_j)=\{\mu_q^j(x_j.e_q^j),x_q\},q\in J.$$

Из этих подмножеств выделяются подмножества:

$$M_j=\{\mu_q^j(x_j.e_q^j),x_q\},\ j=1,2,\ldots..,J,q\in J.$$

Тогда степень достижения каждой цели в рамках рассматриваемой диаграммы причинно-следственных связей определится из выражения:

$$\lambda_j=\max\{M_j\}=\max\{\mu_q^j(x_j.e_q^j),x_q\},\ j=\overline{1.J},q\in J.$$

Структуру причинно-следственных взаимодействий целей управления ВП в виде ориентированного графа связей, показанную на рис. 1.13, можно представить в виде графа G, для которого имеем:

$$X=\{x_1,x_2,x_3,x_4,x_5,\},$$
$$E=\{e_1,e_2,e_3,e_4\},$$
$$e_1=\{x_2,x_4,x_5\},$$
$$e_2=\{x_1,x_5\},$$
$$e_3=\{x_1,x_3,x_4\},$$
$$e_4=\{x_4,x_5\}.$$

Геометрическая интерпретация гиперграфа $H=(X,E)$ показана на рис. 1.14.

Для рассмотренного гиперграфа H определяется как:

$$B(H)=\langle F,X,E\rangle,$$
$$X=\{x_1,x_2,x_3,x_4,x_5,\},$$

$$E = \{e_1, e_2, e_3, e_4\},$$
$$\mathrm{F} = \{(x_1, e_2), (x_1, e_3), (x_2, e_1)(x_3, e_3)(x_4, e_1)(x_4, e_3)(x_4, e_4)(x_5, e_1)(x_5, e_2)(x_5, e_4)\}.$$

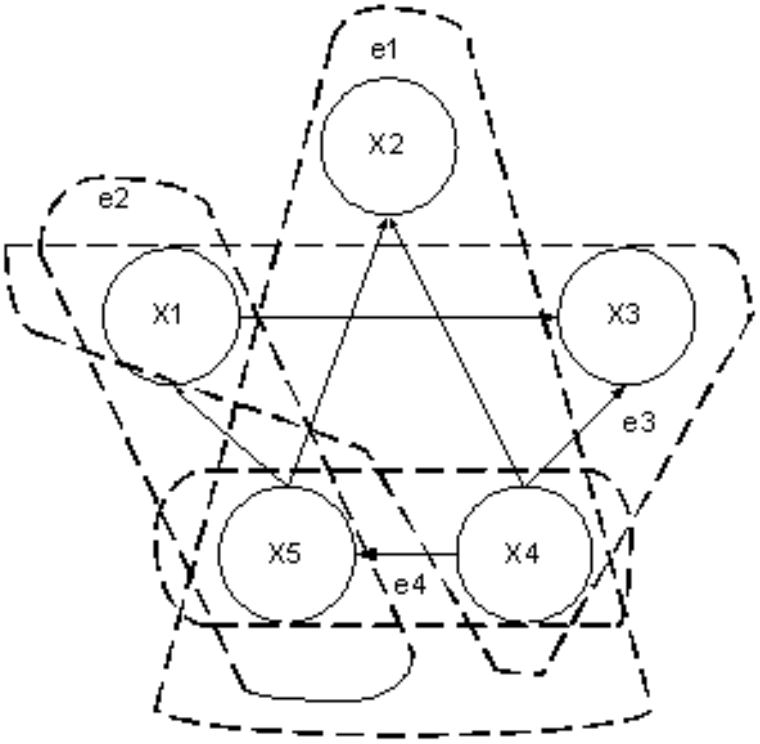


Рис. 1.14. Гиперграф исходного графа

Тогда соответствие $B(H) = \langle F, X, E \rangle$ можно представить графически, как показано на рис. 1.15.

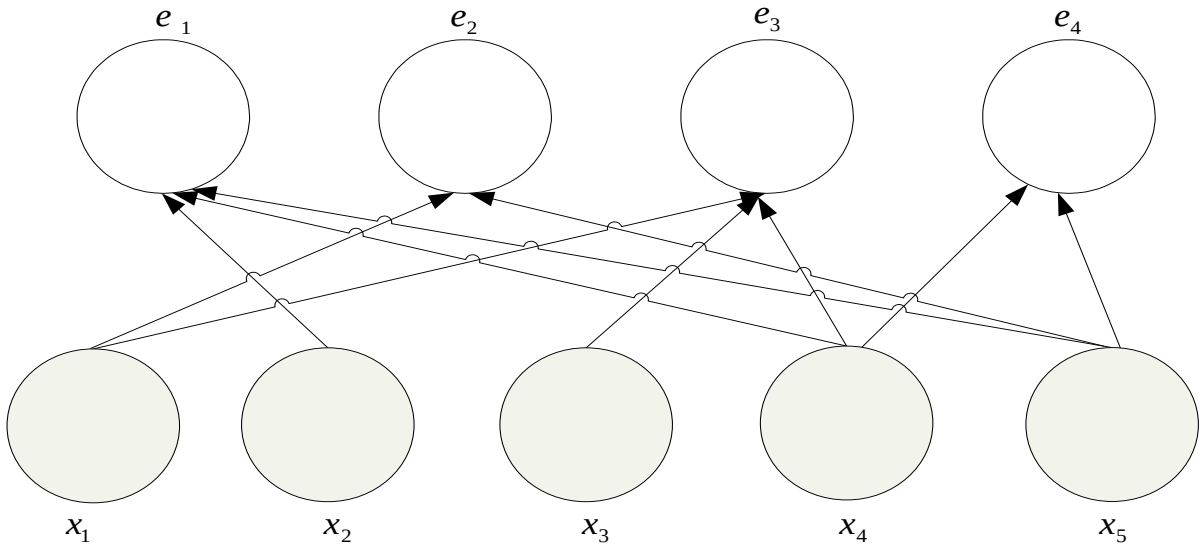


Рис. 1.15. Гиперграф H исходного графа $G = (X, L)$
с учетом глубины раскрытия целей управления ВП

На рис. 1.15 показано, что степень влияния целей различная, что отражено толщиной связей графа, т.е.

$$s_1 = E(x_1) = \{\mu_q^1(x_1.e_q^1), x_1\} = \{(0, x_1), (0.6, x_3)\};$$
$$s_2 = E(x_2) = \{\mu_q^2(x_2.e_q^2), x_2\} = \{(0, x_2)\};$$
$$s_3 = E(x_3) = \{\mu_q^3(x_3.e_q^3), x_3\} = \{(0, x_3)\};$$
$$s_4 = E(x_4) = \{\mu_q^4(x_4.e_q^4), x_4\} = \{(1, x_2), (0.2, x_3)(0.6, x_5)\};$$
$$s_5 = E(x_5) = \{\mu_q^5(x_5.e_q^5), x_5\} = \{(0, x_5), (1, x_2)(0.4, x_1)\}.$$

Тогда

$$\lambda_1 = \max\{M_1\} = 0.6;$$
$$\lambda_2 = \max\{M_2\} = 0;$$
$$\lambda_3 = \max\{M_3\} = 0;$$
$$\lambda_4 = \max\{M_4\} = 1;$$
$$\lambda_5 = \max\{M_5\} = 1.$$

Следовательно, кортеж целей с точки зрения первоочередности их достижения можно записать в следующем виде:

$$(x_4 \equiv x_5) \succ x_1 \succ (x_2 \equiv x_3).$$

Таким образом, можно определить приоритеты задач управления и наилучшим образом удовлетворить на насыщенном рынке постоянно изменяющиеся потребности, создавая для ВП сильные конкурентные преимущества.

Литература:

1. Еникеева Л.А., Соколовская С,А. Методы управления виртуальными предприятиями: монография. – СПб.: СПбГИЭУ, 2010. – 121 с.
2. Моделирование и безопасность виртуальных предприятий. // Безопасность современных информационных технологий: монография. / под общей ред. Е.В. Стельмашонок. – СПб.: СПбГИЭУ, 2012. – С. 347-367.
3. Сердюк В.А. Сетевые и виртуальные организации: состояние, перспективы развития // Менеджмент в России и за рубежом. - №5, 2002 [электронный ресурс]. URL: http://www.mevriz.ru/articles/2002/5/1033.html (дата обращения 11.11.2017).

1.5. Основы технологии поисковой оптимизации сайта для обеспечения его продвижения и защиты информации

Еникеева Л.А., Торосян Е.К.

В современных условиях развития информационно-коммуникационных технологий эффективное продвижение продуктов и услуг представляется практически невозможным без наличия у компании собственного веб сайта для охвата многомиллионной целевой аудитории.

Таким образом, компании заинтересованы в присутствии в Интернет пространстве и активном развитии в этом направлении, а для этого требуются определенные знания и навыки. Активно меняются приоритеты, поэтому ставка делается на интернет-маркетинг и традиционно каждая компания имеет свой сайт.

С активным развитием поисковых систем и увеличением доли интернет-коммерции становятся востребованными специальные методы поисковой оптимизации и продвижения сайтов компаний, которые дают возможность им быть конкурентоспособными на рынке. Основной задачей поисковой оптимизации является выведение сайта в первую десятку результатов выдачи запросов, расположенных на первой странице выдачи [1, 3, 7]. Целью данного исследования является анализ методов и инструментальных средств оптимизации сайтов для повышения эффективности ранжирования сайтов в современных поисковых системах.

Рассмотрим основные схемы работы поисковых систем. Поиск и обработка информации в сети происходит по следующим параметрам: сбор информации со страниц сайтов в сети Интернет; индексация сайтов; поиск по запросу; ранжирование результатов. Схема работы поисковых систем представлена на рис. 1.16.

Краулеры – поисковые роботы, которые постоянно сканируют новые документы и составляют номенклатуру документов для следующего обхода другим роботом, который загружает информацию со страницы. Затем система проверяет, имеется ли документ именно в этом виде уже в поисковом индексе. При обнаружении документа работа с ним прекращается. Если документа нет или он был редактирован с последнего момента посещения роботом, то процесс работы с данным документом продолжается и документ проверяется на вирусы и спам.

В случае если документ проходит всю проверку, то далее роботом-индексатором производится лексический анализ:

1. выбор структурной единицы документа;
2. выбор структурной единицы документа;
3. выбор структурной единицы документа;
4. нормализация лексем (токенов);
5. приведение к общей основе словоформ и производных форм.

Например, при поиске запроса «R.U.S.», то поисковик устроит и написание «RUS». Следовательно, эти два токена должны быть нормализованы и объединены в классы эквивалентности.

Важной составляющей работы поисковых систем Google и Яндекс является использование системы кластеров, когда информация разделяется на определенные области (кластеры).

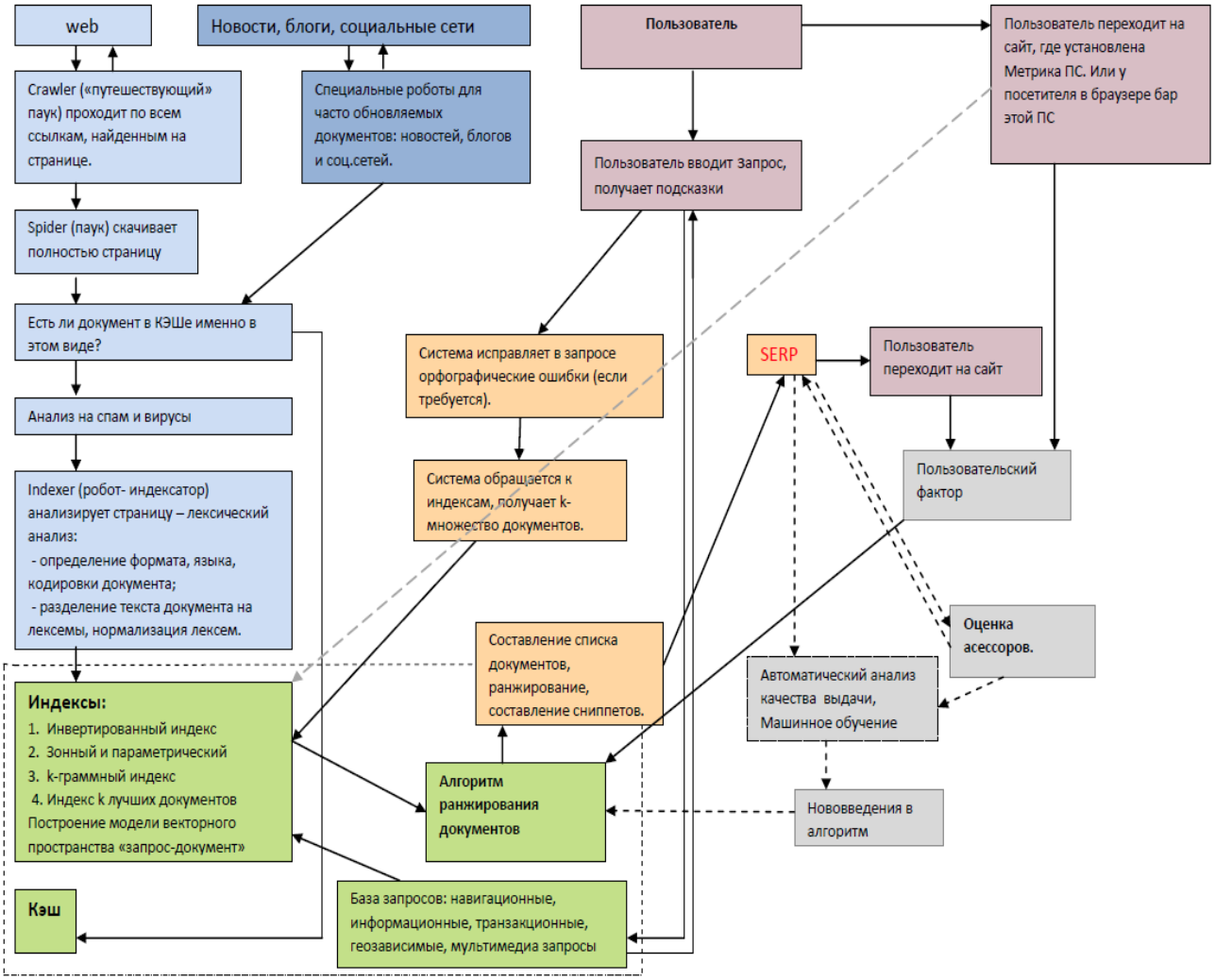


Рис. 1.16. Схема работы поисковых систем

Роботы-сканеры выполняют индексацию сайта с целью получения данных о размещенной на них информации. Существуют два вида сканирующих роботов: основной робот-сканер и робот сканер, который отвечает за сбор информации на ресурсах с регулярным обновлением контента. Второй тип робота-сканера необходим для быстрого обновления списка и значения их индексов проиндексированных сайтов в поисковой системе. Для более полного обеспечения сбора информации в поисковой системе Яндекс используются обновления базы поиска и обновления программного кода.

Компания Яндекс хранит в секрете IP адреса поисковых роботов. Но в логах некоторых сайтов можно увидеть текстовые заметки, оставленные поисковыми роботами.

Продвигать сайты в Google, особенно на начальном этапе, немного сложнее, чем на Яндекс. Продвижение молодого сайта в Google затруднительна, так как на новые веб-ресурсы накладывается фильтр (так называемая «песочница»). Google при ранжировании использует порядка 200 факторов, а оптимизатор может повлиять лишь на некоторые [6, 7].

С другой стороны, информация, только что размещенная на сайте, может в *считанные* минуты попасть в основную выдачу. Поисковые роботы Google в три раза быстрее, чем роботы других поисковых систем. Фильтры (критерии «нормальности» сайта) почти не меняются с момента начала их внедрения.

Факторы, влияющие на принятии решения поисковой машиной о включении страницы в поисковую выдачу и при определении степени ее релевантности тому или иному запросу пользователя, можно разделить на *внешние и внутренние факторы.*

Внешние факторы – это параметры, на которые оптимизатору влиять удается далеко не всегда. Данные действия в большинстве случаев связаны с финансовыми затратами. Внутренние факторы напротив, находятся в полном ведении оптимизатора и при наличии знаний и опыта могут быть настроены соответствующим образом. Однако ни одна поисковая система в явном виде не указывает все факторы, которые используются в алгоритмах.

Степень влияния основных факторов на ранжирование сайта:

- высокая:
- текстовое содержание страницы,
- мета-теги,
- плотность ключевых слов,
- наличие ссылочной массы,
- поведенческие факторы,
- возраст сайта от 6 месяцев;
- средняя:
- наличие грамотной перелинковки,
- уровень доверия поисковой системы к ресурсу;
- низкая:
- ТИЦ и PageRank.

Например, основные факторы ранжирования, используемые системами Яндекса и Google рассмотрены в [3].

Яндекс:

- поведенческие факторы;
- ссылочная масса;
- регион;
- тематика;
- возраст сайта;
- оптимизированный текст;
- наличие контактной информации.

Google:

- социальная значимость;
- актуальность контента;

- скорость загрузки страниц сайта;
- наличие ключевого слова в домене;
- количество и авторитетность ссылок с уникальных доменов на сайт;
- факторы бренда.

Высокая оценка поисковых систем присваивается только тем проектам, которые имеют авторитетную поведенческую и конверсионную метрику. Продавать можно все, главное – правильно подойти к решению организационных вопросов, грамотно подобрать маркетинговые стратегии, которые помогут в достижении поставленных целей. Сайт – это, всего лишь, средство с помощью которого можно рекламировать любой товар или услугу.

Рассмотрим подробнее саму технологию поисковой оптимизации сайта.

Поисковая оптимизация предусматривает пошаговое выполнение комплекса мероприятий, которые в результате смогут повысить позиции сетевого проекта в поисковой выдаче:

1. Выбор стратегии оптимизации сайта.
2. Выбор технологии составления семантического ядра сайта
3. Выбор способа внутренней оптимизации сайтов
4. Способы внешней оптимизации сайтов
5. Анализ и учет поведенческих факторов
6. Подбор инструментальных средств оптимизации сайта

Именно благодаря произведению этих действий сайт повышается в рейтинге поисковых систем.

Первоначально сайт проверяется на аффилиты, то есть на схожесть сайта, со схожими адресами и контентом, а также осуществляется настройка и при необходимости редирект сайта (например, подробнее описано в [2]: 7162.kz на www.7162.kz.) Таким образом, сайт приобретает больший уровень доверия, чем прежде, что позитивно сказывается на его местоположении при вводе ключевых слов.

Контент также играет большую роль при оптимизации сайта, а также для его продвижения. Соответственно очень важно проверить сайт на качество контента и на копирайтинг, при проведении проверки может быть выявлено, что тексты на сайте требуют обработки и доработки. Это необходимо из-за схожести текстов, размещенных на аналогичных сайтах конкурентов. Данную похожесть поисковые машины оценивают очень низко и присваивают плохой индекс, и как следствие сайт теряет свои места на выходе.

Система администрирования сайта позволяет редактировать тексты, контактную информацию, работать с разделами и пунктами меню (добавление, редактирование, удаление). На главной странице должен

присутствовать баннер-ротатор со сменными изображениями. Сайт должен быть разработан в соответствии с техническим требованиями, могут быть дописаны модули, облегчающие работы по поисковой оптимизации.

Если сайт работает на CMS DLE, который в стандартном пакете не совсем понятен для поисковых машин и использует базовые модули DLE, то это не является доверительным у поисковых машин, что снижает его индекс при ранжировании в поисковой выдаче. Так же изначально если на сайте не находилась статистика посещаемости, то следует добавить и настроить счетчики в Google Analytics и Яндекс Метрике. Данные сервисы помогают как оптимизаторам, так и маркетологам иметь представление о посещаемости сайта, о количестве переходов, о глубине проникновения на сайт и т.д.

Таким образом, главная цель оптимизации сайта: генерация потока новых клиентов и их удержание.

Если, поисковая оптимизация сайта пройдет успешно, сайт займет лидирующие позиции в поисковой системе Google и, как следствие, на сайт попадет большее количество пользователей и потенциальных клиентов, чем до начала продвижения. Приблизительно 30-50 пользователей ежедневно будут оценивать «7162.kz» по их представительству в сети Интернет [2, 5, 6]. При устаревшем дизайне, неудобной навигации, пользователи не будут совершать полезные действия: звонок, отправка сообщения, участия в опросах. Поэтому, внешний вид и удобство сайта не менее важны, чем его продвижение.

Следующим пунктом выполнения работы с сайтом является настройка/создание файла sitemap.xml (карта сайта). Создание этого файла, как правило, необходимо для сайтов с большим количеством страниц (100-150), но и для небольшого сайта она будет весьма полезна. Поисковая система не узнает о существовании сайта автоматически, чтобы сайт добавить в свою базу (индекс) и в дальнейшем его показывать, в поисковике есть специальные роботы, которые заходят на сайты и по определенным алгоритмам просматривают все страницы, изучают их и заносят в базу. Существует вероятность ошибки робота, и он может не зайти на сайт, и соответственно, не внести его в базу и именно для исключения этой ошибки создается карта сайта.

Так же сайту был присвоен регион, чтобы так же облегчить поиск для пользователей и поднять сайт при введении таких ключевых слов как «услуги», «гостиницы» [2]. Так пользователи из этого региона или области сразу будут видеть в выдаче сайт. Для каждой страницы быть прописаны title, уникальные для каждой страницы заголовки.

Помимо этого, при оптимизации сайта создавались расширенные *сниппеты* - описание сайта в выдаче поисковых систем. То есть в результате поиска

все сайты пронумерованы, потом идет заголовок, и далее идет само описание сайта, таким образом, важно чтобы это описание сайта было корректным и максимально полным. В сниппете обязательно должен быть указан адрес, телефон и время работы.

Для повышения привлекательности сайта в поисковых системах и для увеличения возможных вариантов перехода на сайт, должны быть прописаны alt и title у изображений на продвигаемых страницах. Alt помогает Google, а с недавнего времени и Яндекс индексировать изображения на сайте, для того чтобы изображение с сайта попадало в Яндекс картинки (и Google соответственно). Это увеличивает трафик и добавляет информацию поисковым системам о том, что на сайте есть что-то полезное, необходимое для пользователя.

В процессе оптимизации сайта была создана перелинковка с одних страниц сайтов на другие страницы, которые имеют больший вес для продвижения сайта в поисковых системах. Создание таких ссылок способствует повышению индекса сайта, и ссылочная масса, внешняя и внутренняя, оказывает большое влияние на рейтинг сайта. Данный процесс так же заменяет покупку ссылок на других ресурсах, что значительно уменьшает издержки на продвижение сайта.

Так же рекомендуется добавление *фавикона* – иконки, которая появляется в поисковой выдаче справа от описания сайта, которая дает возможность заметить сайт сразу при выведении списка. Как правило, эта иконка является логотипом компании, и может выглядеть следующим образом, рис. 1.17.

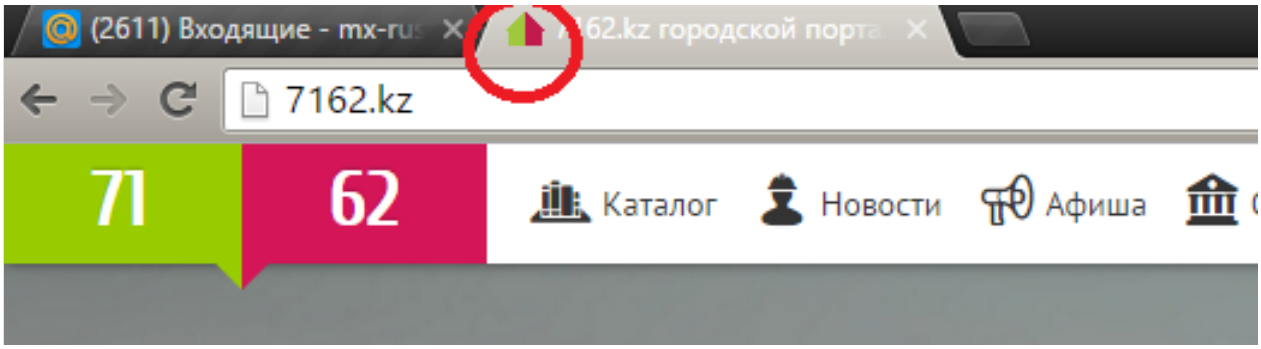


Рис. 1.17. Пример фавикона [2]

Для сайта также должна быть предусмотрена настройка 404 страницы ошибки. Ошибка 404 – это ошибка о том, что страницы, на которую пытается перейти пользователь, не существует. Зачастую данная страница пользователей пугает, при этом фактически пользователь уходит с сайта и видит заставку с белым фоном и крупными словами об ошибке, рис. 1.18. Обработка ошибки означает, что пользователь не уйдет с сайта, вместо текста ему покажут информацию о том, что такой страницы не существует.

Рис. 1.18. Страница, которая выводит ошибку 404

И конечно, самый важный фактор, который влияет на ранжирование сайтов в поисковых системах – это ссылочная масса. Безусловно, необходимо провести работу со ссылочной массой, увеличить ее, и как следствие увеличится количество переходов на сайт, и его позиция при поисковой выдаче.

В перспективе, в процессе работы над SEO оптимизацией сайта планируется производить дополнительные корректировки всех тэгов (title, description, h1,h2) в зависимости от результатов выдачи, наполнение сайта более новым контентом и дальнейшее увеличение внешней ссылочной массы для SEO оптимизации сайта [4-6]. В перспективе со временем стоит подумать о ребрендинге сайта, то есть стоит разработать абсолютно новый сайт, который будет больше соответствовать стандартам, станет еще более простым для использования, более современным в дизайне, где могут использоваться более новые технологии написания, для повышения его привлекательности для поисковых систем и клиентов.

Рассмотрим основные инструментальные средства, используемые при оптимизации сайтов. Все инструменты делятся в зависимости от их использования при определенных этапах продвижения:

1. Аудит сайта (audit.megaindex.ru).
2. Внутренняя оптимизация сайта (http://wordstat.yandex.ru; https://adwords.google.com/o/KeywordTool; http://www.content-watch.ru/).
3. Внешняя оптимизация сайта (Яндекс.Вебмастер; Google.Webmaste; Плагин Wink'a Sape).
4. Аналитика и отслеживание показателей продвижения (http://www.allpositions.ru; Яндекс. Метрика; Google. Analytics).

Audit.megaindex.ru – это новейшая разработка компании ALTWeb-Group, предоставляющая возможность изучить детали работы сайта и получить информацию об оптимизации собственных ресурсов, а также данные об оптимизации за счет внешних факторов (рис. 1.19).

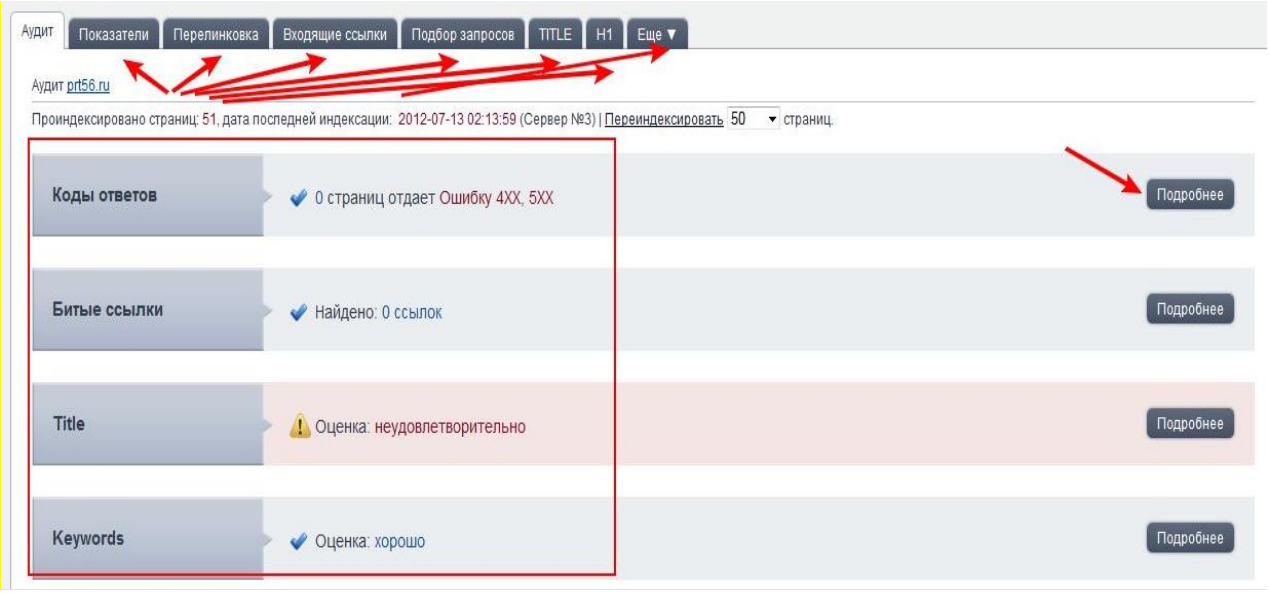


Рис. 1.19. Пример использования интерфейса Megaindex –аудит

Анализ релевантности страницы определенному запросу (megaindex.ru) – сервис проанализирует соответствие документа запросу с позиции поисковых систем и выдаст результат. Как видно, у проверяемой страницы по всем пунктам найдено стопроцентное соответствие (рис. 1.20).

Яндекс.Wordstat – инструмент от поисковой системы Яндекс, позволяющий узнать статистику заданных ключевых слов. Находится по адресу: http://wordstat.yandex.ru. Пример статистики ключевого слова «стенд» указан на рис. 1.21.

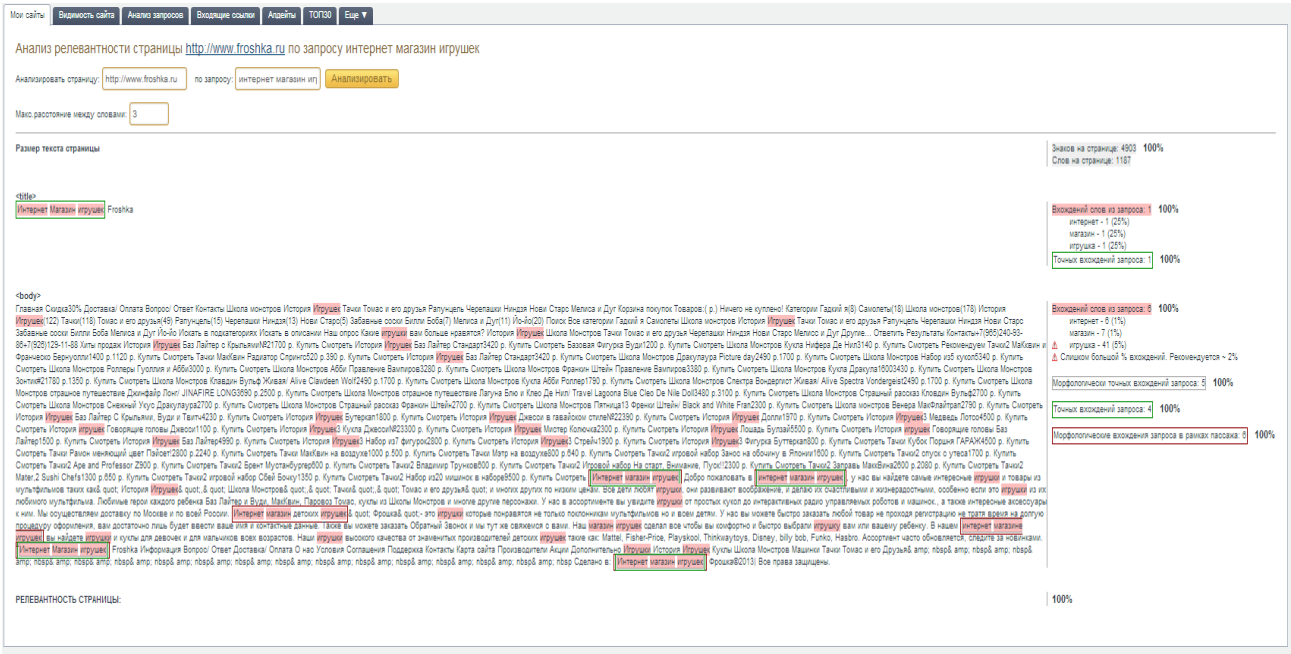


Рис. 1.20. Инструмент анализа релевантности страницы MegaIndex

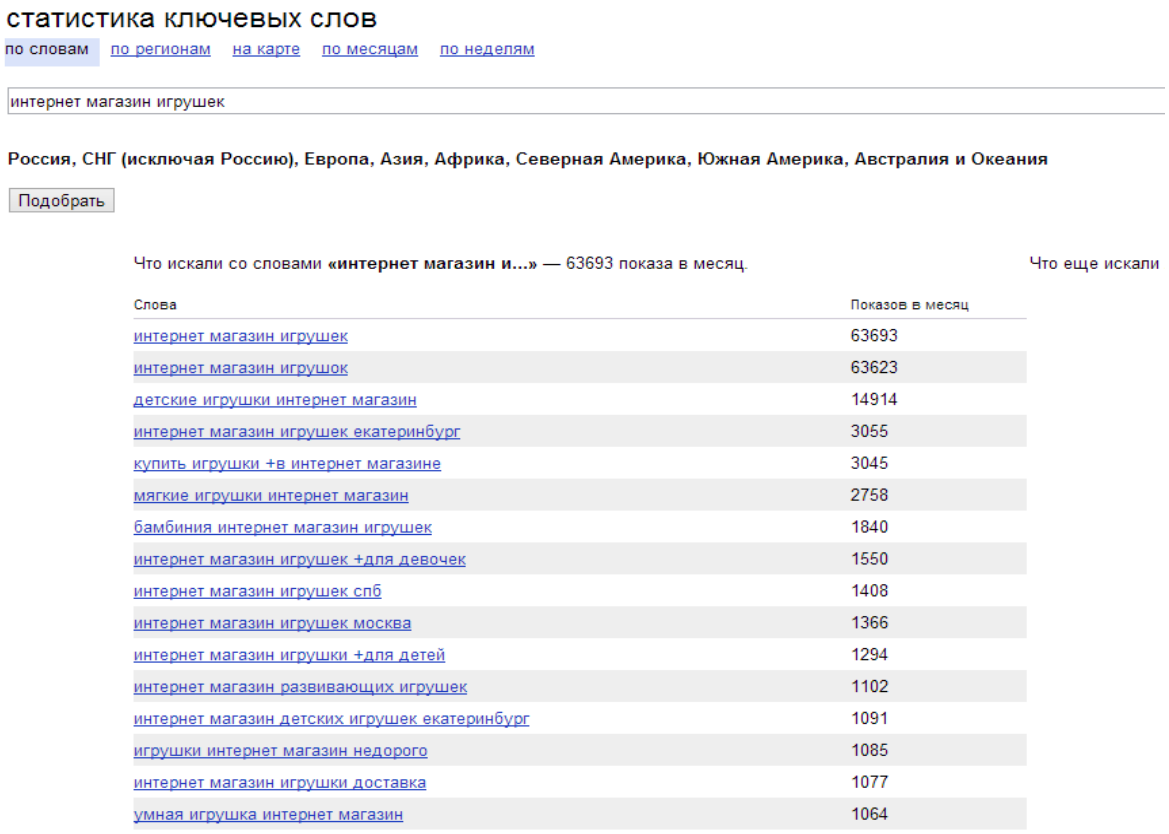


Рис. 1.21. Пример использования Яндекс.Wordstat

Google Keyword Tool – аналогичен Яндекс.Wordstat, но функции шире: помимо данных о самих ключевых словах, получаем оценку конкурентности, количество показов в месяц для всего мира и для заданного региона. Инструмент находится по адресу: https://adwords.google.com/o/KeywordTool. Пример пользования сервисом приведен на рис. 1.22.

Анализ текста на плагиат (http://www.content-watch.ru/) – сервис проверки текста или страницы сайта на уникальность, результатом анализа является процентное количество уникального текста, а также список HTML-страниц других сайтов, которые включают в себя части проверяемого текста.

Инструменты для внешней оптимизации: Яндекс.Вебмастер и Google.Webmaster – два основных инструмента, которые могут пригодиться абсолютно всем. Важно отслеживать положение сайта сразу для обоих основных поисковых систем для защиты от различных неприятных неожиданностей и внося своевременные коррективы в индексацию сайта и Яндексом и Google.

Данные инструменты помогают:

- Отслеживать количество индексируемых страниц;
- Добавлять новые страницы для индексации, в том числе, карты сайта, в специальных формах для добавления;

– Настраивать параметры сайта: выбор главного зеркала, проверка robots.txt

– Производить анализ входящих внутренних и внешних ссылок;

– Отслеживать недоступные «битые» ссылки на сайте;

– Анализировать популярные поисковые запросы, по которым приходят на сайт;

– Настроить географию сайта (регион, адреса и телефоны);

– Анализировать сайт на наличие вирусного кода;

– Проверить сайт на доступность загрузки страниц и многое другое.

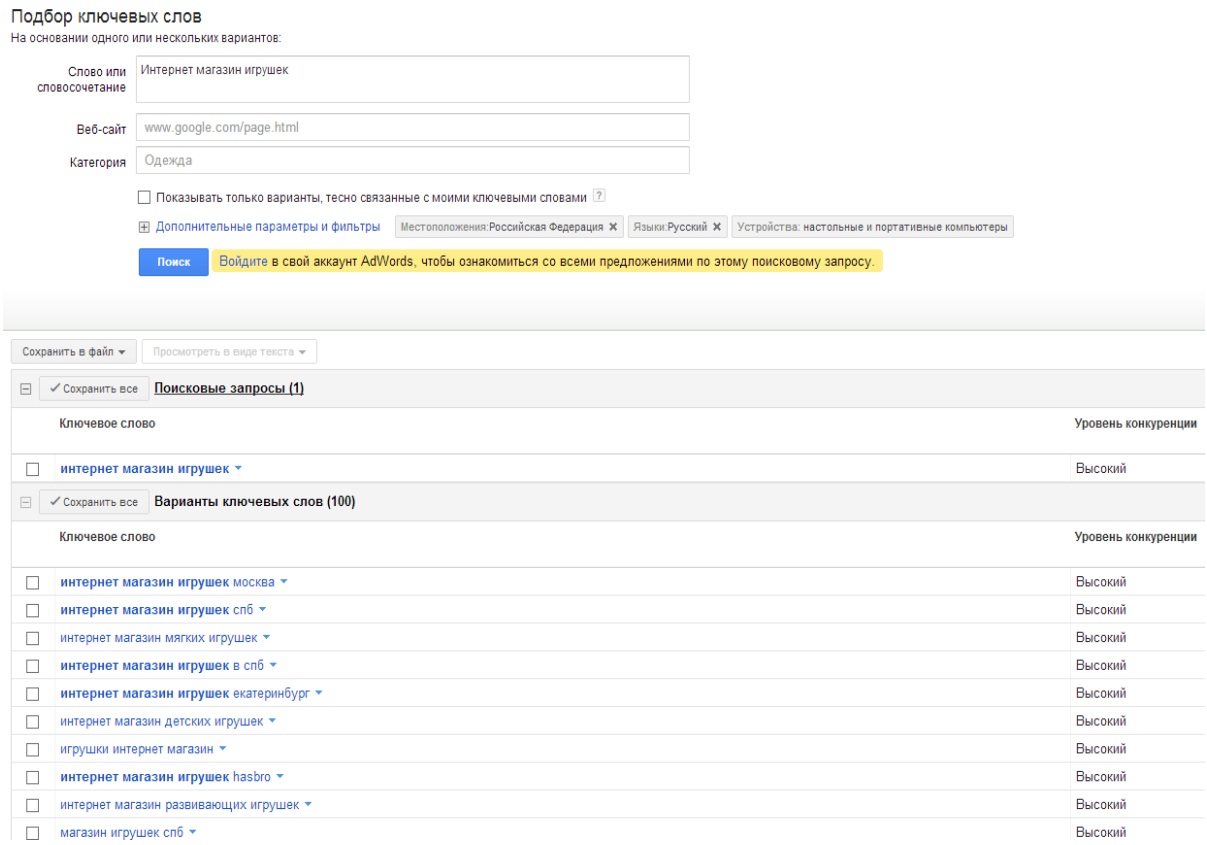


Рис. 1.22. Пример использования Google Keyword Tool

Плагин Wink'a Sape – удобное средство, которое значительно облегчает работу с покупкой ссылок и помогает сэкономить бюджет на продвижении сайтов клиентов, а также экономит очень много времени при фильтрации заявок на сайты.

Инструменты для аналитики и отслеживания показателей сайта. Яндекс. Метрика и *Google. Analytics* – бесплатные сервисы, предоставляемые для создания детальной статистики посетителей веб-сайтов. Сервисами контролируются такие параметры как переходы на внутренние страницы и время выхода с сайта в случае неудачного поиска, тенденции поведения пользователей и среднее время их пребывания на сайте, количество

трафика из поисковых систем и ссылающихся сайтов, статистику кликов на ссылки определенной страницы и т.д.

Есть множество сервисов для анализа позиций сайта, но по адресу: http://www.allpositions.ru расположен один из лучших сервисов, позволяющий определить позиции в выбранной поисковой системе и в выбранном городе/стране. Присутствует гибкая система настройки времени проверки позиций и генерация понятных отчетов изменений позиций. Интерфейс сайта для мониторинга позиций указан на рис. 1.23.

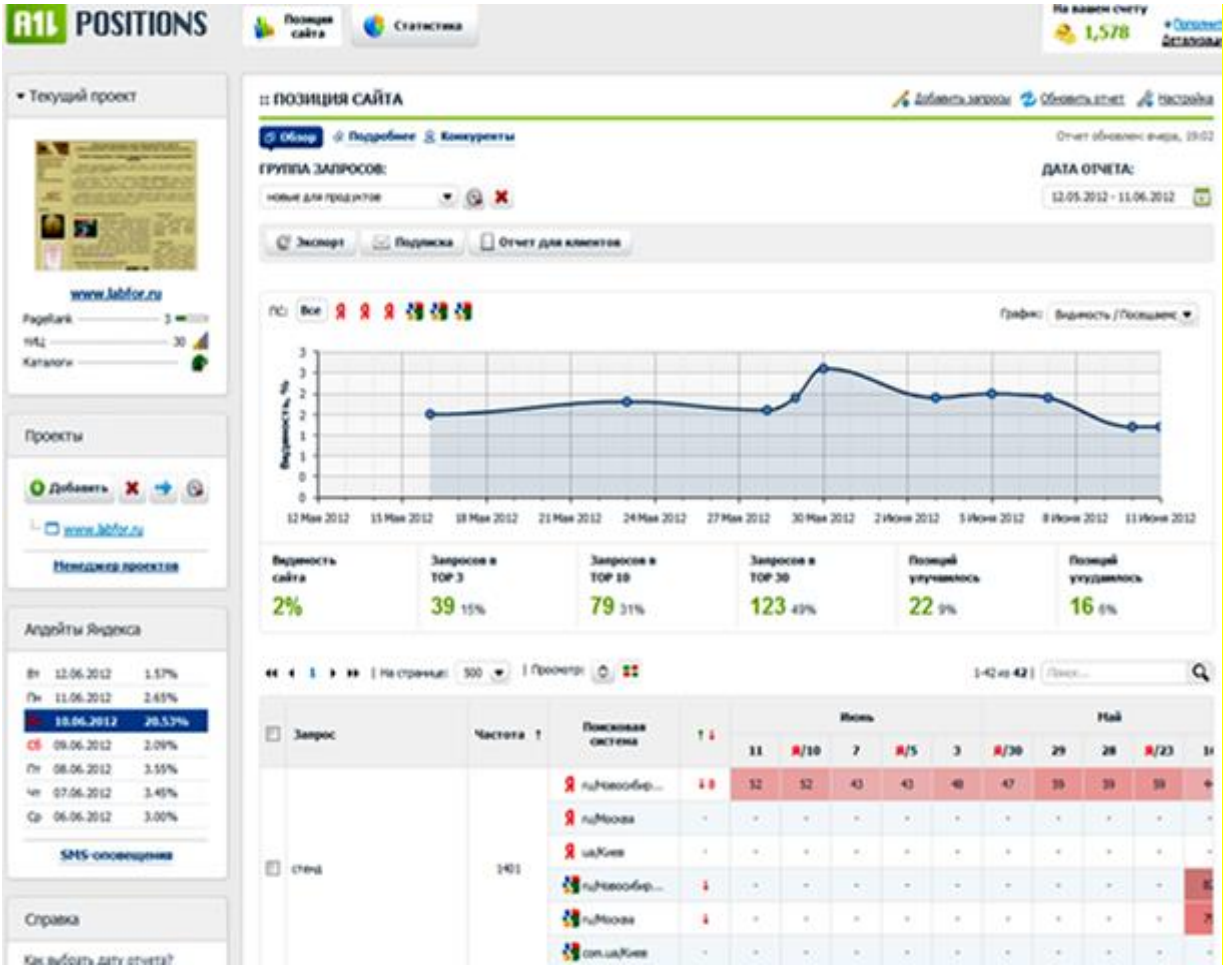


Рис. 1.23. Сервис мониторинга позиций сайта AllPosition

Результаты применения поисковой оптимизации сайта

С помощью инструмента Google Analytics можно следить за показателями и статистикой сайта (рис. 1.24).

Также можно увидеть, какова посещаемость сайта (рис. 1.25), какая целевая аудитория, какие страницы более посещаемые, поисковые фразы по которым нас находят, процент отказов, время и глубина просмотра на сайте, и много другой полезной информации, которая может помочь в оптимизации сайта.

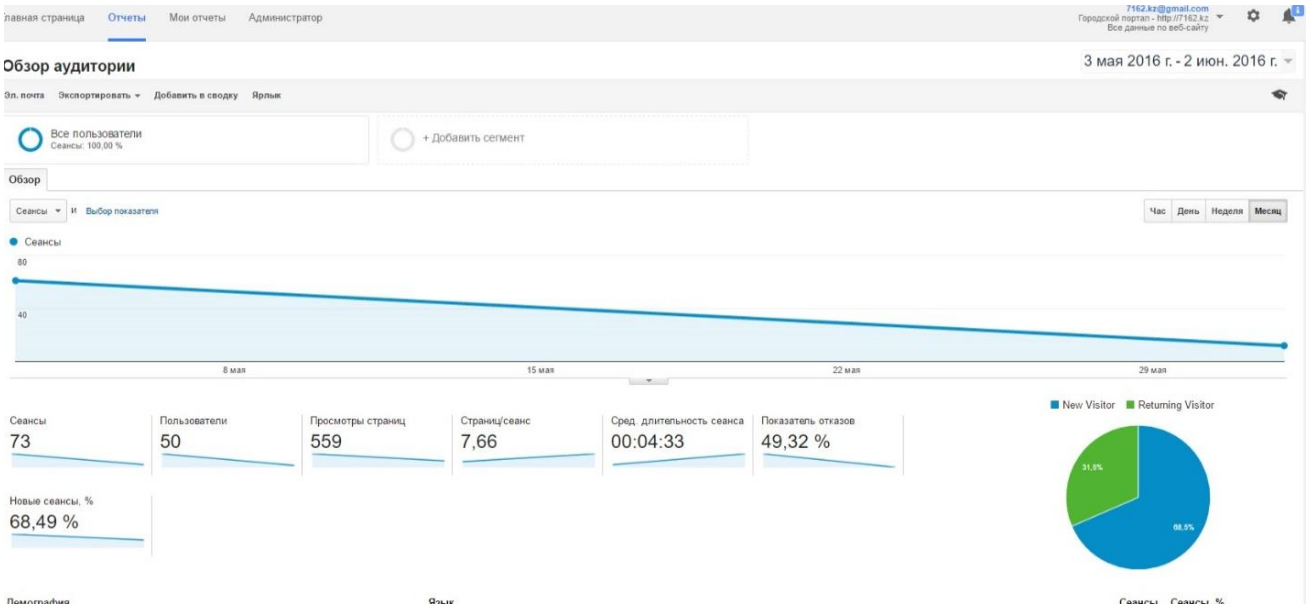


Рис. 1.24. Использование Google Analytics

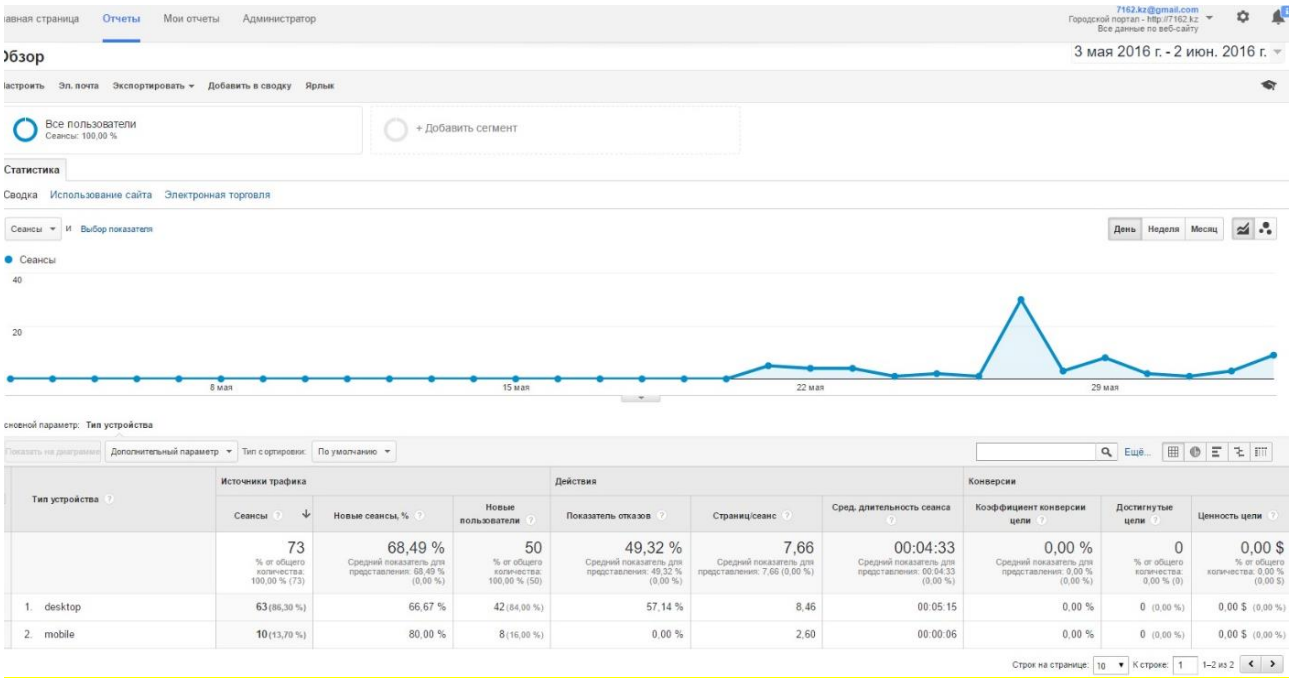


Рис. 1.25. Статистика входа

Следить за позициями запросов в поисковых системах можно через сервис AllPosition (рис. 1.26). Сервис предоставляет отчет по позициям в поисковых системах Яндекс и Google по всем заданным ключам. В отчете мы видим, что 80% ключей ранжируется в ТОП 10, а большая часть в ТОП 3. Из этого следует что использованная нами методика оптимизации, повысила уровень ранжирования сайта «7162.kz» в поисковых системах на долгосрочный период [2].

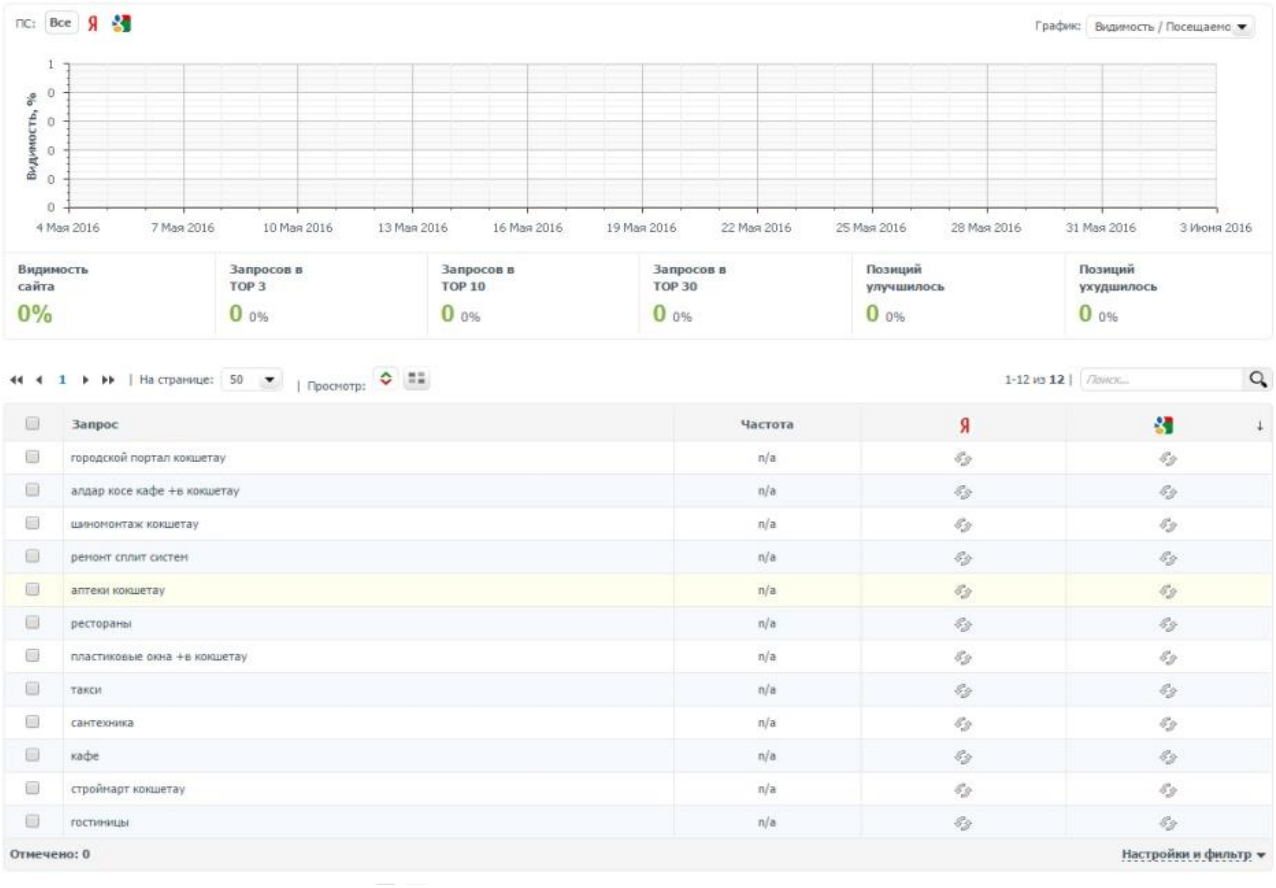


Рис. 1.26. Позиции запросов в поисковых системах

Рекомендуемая методика сможет улучшить ранжирование сайта и вывести большую часть запросов в ТОП 10 поисковых систем на долгосрочный период, что в следствии приведет к постепенному увеличению его посещаемости.

Данные методы будут полезны всем, кто занимается коммерческой деятельностью в Интернете, для привлечения аудитории из поисковых систем и повешения продаж через интернет.

Литература:

1. База статей по поисковому маркетингу, Продвижение сайта. Профессиональные советы экспертов [электронный ресурс]. URL: http://www.optimization.ru/subscribe (дата обращения 11.11.2017).
2. Торосян Е., Хамзин Р. Поисковая оптимизация как инструмент развития бизнеса // Сборник материалов II-ой международной научно-практической конференции. Национальная академия образования им. И. Алтынсарина, Уральский государственный педагогический университет. 2015. – Екатеринбург, УГПУ, 2015. – С.557-559.
3. Яндекс Help [электронный ресурс] URL: http://help.yandex.ru/webmaster/ (дата обращения 11.11.2017).

4. Bas van den Beld, «SEO For Europe Is More Than Just Using Different Languages», Sep 15, 2009.
5. Beel, Jöran and Gipp, Bela and Wilde. Academic Search Engine Optimization (ASEO): Optimizing Scholarly Literature for Google Scholar and Co. [Электронный ресурс] URL: https://docear.org/papers/ Academic%20Search%20Engine%20Optimization%20(ASEO)%20--%20preprint.pdf (дата обращения 11.11.2017).
6. George S. Spais (Greece), Search Engine Optimization (SEO) as a dynamic online promotion technique: the implications of activity theory for promotion managers, Innovative Marketing, Volume 6, Issue 1, 2010. – P. 7-24.
7. Google Support [Электронный ресурс]. URL: http://support.google.com/webmasters/ (дата обращения 11.11.2017).
8. (8)SEO Энциклопедия [Электронный ресурс]. URL: http://www.webeffector.ru/wiki/ (дата обращения 11.11.2017).

1.5. Вопросы безопасности в Интернете вещей

Полегенько А.М

Интернет вещей (Internet of Things, IoT) – концепция, предполагающая объедение в сеть устройств (вещей), способных взаимодействовать друг с другом на основе встроенных технологий, которые поддерживаются данной сетью. Устройствами (вещами) являются «умные» гаджеты, «умная» техника и другие сетевые устройства, которые могут быть использованы в обиходе человека или его дома. Безопасность Интернета вещей становится ключевым аспектом при построении таких сетей. Получив доступ к одному устройству, злоумышленник может проникнуть в сеть, и тогда уже угрозам подвергается любая конфиденциальная информация. Отсюда вытекает актуальность вопросов безопасности информации в подобных сетях, где необходимо учитывать ограничения устройств, входящих в них.

Понятие «Интернет вещей» было предложено Кевином Эштоном, в Массачусетском технологическом институте в 1999 году. С тех пор концепция начала свое стремительное развитие и положила начало для многих стартапов.

Сегодня нас окружает все большее количество гаджетов, способных обмениваться друг с другом данными с участием пользователя или без него. Объединяясь в сеть, различные устройства от фитнес-трекеров до дистанционной системы управления электроснабжением дома, обрабатывают и передают информацию, относящуюся к пользователю. Это может быть личная информация, данные о настройках устройств и сети или даже конфиденциальная финансовая информация.

Структуру Интернета вещей в общем случае можно представить как совокупность следующих элементов:

- непосредственно сами «вещи», то есть устройства, датчики и сенсоры, физические объекты, которые в привычном понимании не предназначались для подключения к сети. Такие устройства должны быть однозначно идентифицированы с помощью программно-аппаратных средств – это могут быть RFID-метки, штрих-коды, MAC-адреса и др.;
- сеть – вариации проводных и беспроводных сетей, поддерживающих разные протоколы и стандарты и построенных с помощью маршрутизаторов и шлюзов;
- центры обработки данных – хранилища и вычислительные ресурсы, задействованные в сборе, анализе и обработке данных «сети вещей», например, это могут быть «облака» или «туманные узлы».

Таким образом, подходы к построению системы безопасности должны рассматривать каждый из структурных элементов и еще решать проблемы, вытекающие при объединении нескольких устройств и создании сети.

Исходя из масштабов сети, выделяют 4 уровня Интернета вещей [2]:

- 1 уровень включает отдельные объекты – «вещи»;
- 2 уровень предполагает создание сети «вещей» на уровне отдельных потребителей, объединяя устройства личного пользования (например, смарт-дом);
- 3 уровень охватывает жизнь целых городов, т.е. подразумевает, например, концепцию создания смарт-городов;
- 4 уровень предполагает объединение всего мира посредством Интернета вещей.

Соответственно, можно говорить о возможности масштабирования сетей подобного рода.

«Вещами» сегодня являются не только предметы личного пользования обычных потребителей, но и различная техника, активно применяемая во множестве сфер деятельности – торговле, транспорте, медицине, строительстве, банкинге, спорте и др. Отсюда следует, что Интернет вещей чаще всего представляет собой гетерогенную сеть, т.е. устройства различных классов и видов объединяются и взаимодействуют между собой.

Кроме неоднородности сетей, особенностью Интернета вещей так же является то, что устройства обладают неодинаковыми вычислительными ресурсами, пропускной способностью и поддерживают разные технологии и протоколы. Отсутствие единых стандартов и протоколов остается серьезной проблемой при построении сети «вещей». Так же многие «вещи» обладают ограниченными возможностями электропитания и должны поддерживать режимы энергосбережения.

Перечисленные особенности Интернета вещей накладывают ограничения и при построении системы безопасности в такой сети. Привычных методов защиты информации в беспроводных сетях может быть недостаточно или же они не могут быть применимы в связи с ограничениями, которые накладывает сеть Интернета вещей.

Основными методами обеспечения безопасности, как и в традиционных сетях, остаются шифрование, идентификация/ аутентификация, внедрение физических мер безопасности.

Система безопасности должна быть спроектирована так, чтобы предусмотреть защиту для устройств и шлюзов, сети передачи, а также приложений, которые разворачиваются для обеспечения функционирования устройств.

Шифрование является широкоприменяемым, эффективным и достаточно гибким решением для обеспечения конфиденциальности информации и создания системы защиты. Однако любое шифрование, а особенно надежное, требует увеличения производительности и дополнительных вычислительных ресурсов, что является не всегда возможным в условиях Интернета вещей.

Что же касается аутентификации, то исследователями было предложено достаточно большое количество подходов, которые могли быть внедрены для решения проблем безопасности [1, 3]. Одним из распространенных методов является двухфакторная аутентификация. Например, аутентификация на основе одноразовых паролей (OTP). При таком подходе после предоставления идентификационных данных, пользователю или устройству необходимо предъявить еще и одноразовый пароль, сгенерированный центром распределения ключей, тем самым подтверждая свою подлинность. Такой метод не требует от устройств дополнительных вычислительных ресурсов или хранилищ, однако является неприменимым для устройств, которые, например, просто не могут поддерживать возможность ввода полученного одноразового пароля. Такая же проблема актуальна и для метода аутентификации, вторым фактором которого является аппаратный идентификатор.

Другие исследования предлагают использовать при аутентификации концепцию «цифровых воспоминаний», которая решала бы проблему запоминания пользователями сложных паролей. Однако такой метод накладывает ограничения на ресурсы устройств.

Предлагаемые методы так же включают и аутентификацию с применением криптографии на основе эллиптических кривых. Несмотря на то, что в этом случае необходимые базовые параметры эллиптических кривых вычисляются не самими устройствами, после вычисления требуется передача достаточно большого объема данных, что может быть ограничено пропускной способностью сети [3].

Таким образом, различные существующие методы аутентификации являются применимыми для отдельной сети и отдельного класса устройств. Применение единых методов и средств затрудняется отсутствием стандартизации и гетерогенностью подобного рода сетей.

Для обеспечения безопасности Интернета вещей может быть предложен подход, который имеет в своей основе концепцию «профилей безопасности». Это означает, что на каждом из уровней Интернета вещей, описанных выше, должна быть обеспечена безопасность в соответствии с неким набором метрик «профиля безопасности». Так как представленные уровни имеют иерархическую структуру, и каждый более высокий уровень можно рассматривать как совокупность элементов более низких уровней, то и «профили безопасности» будут иметь наследственность, т.е., например, для обеспечения безопасности смарт-дома необходимо обеспечить безопасность каждого из устройств и предусмотреть дополнительные меры по защите взаимодействия между узлами.

Вводимые метрики безопасности должны удовлетворять следующим основным показателям:

1. Конфиденциальность и целостность – наиболее важные свойства информации, обеспечиваемые стандартными методами с учетом особенностей устройств сети.
2. Надежность – данная метрика особенно актуальна для критически важных объектов, где сбои в работе могут привести к серьезным последствиям.
3. Масштабируемость – сеть должна поддерживать возможность расширения без ущерба функциональности или безопасности.
4. Поддержка работоспособности (стабильность) – поддерживаемые протоколы, технологии и приложения должны иметь возможность своевременного обновления и сохранения работоспособности устройств.
5. Обнаружение вторжений – в сети должны быть реализованы меры по своевременному обнаружению атак или любых других попыток нарушения работы.
6. Своевременность – метрики, которые должны выражать способность сети реагировать на происходящие события.

Предлагаемый набор меток (рис. 1.27) является минимальным и может быть дополнен и расширен, в зависимости от рассматриваемых архитектур сети.

Обозначим Интернет вещей 1-го уровня (каждое отдельно взятое устройство) за X_1, 2-го уровня (сеть смарт-дома) – за X_2, 3-го уровня (смарт-город) – за X_3, 4-го уровня (всемирная сеть Интернета вещей) – за X_4. Тогда «профили безопасности» можно обозначить как $П$ и представить в виде совокупности метрик безопасности: $П\ \{a_1, a_2, ..., a_m\}$, где a_1,

a_2, …, a_m – метрики, определяющие необходимые требования безопасности. Таким образом, общую схему наследственности «профилей безопасности» в Интернете вещей можно представить в следующем виде:

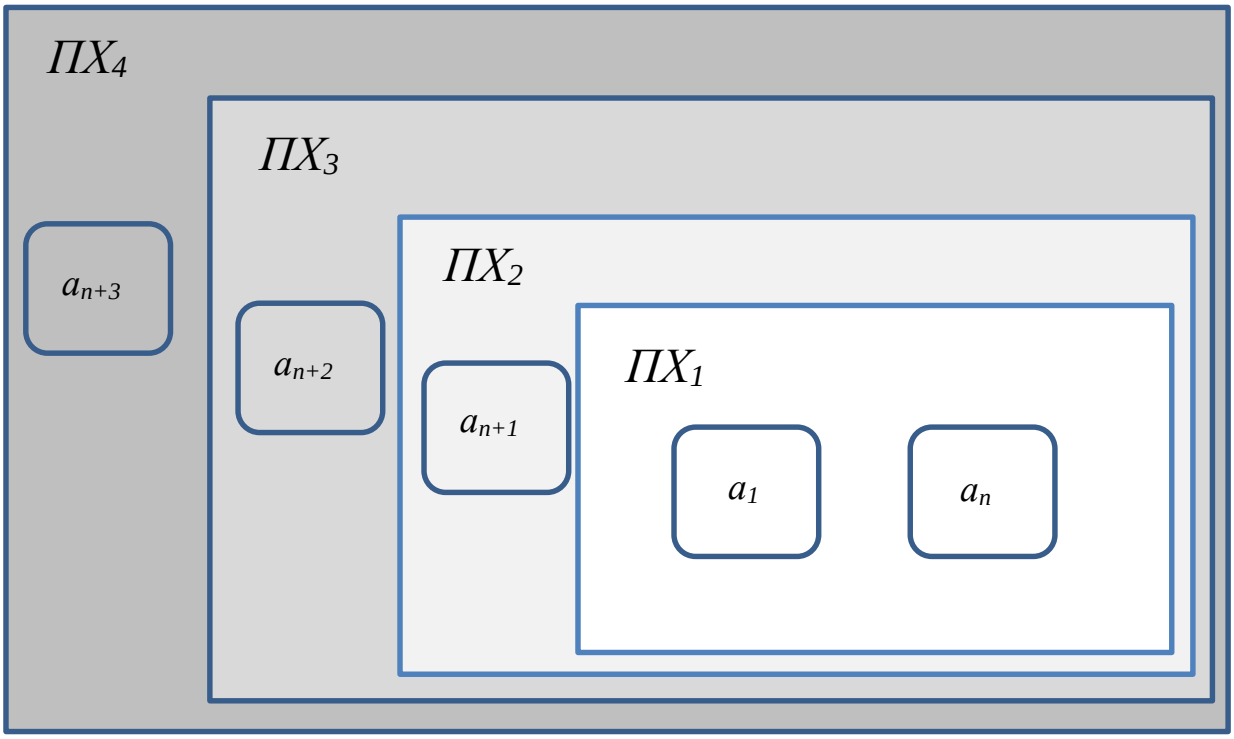


Рис. 1.27. Профили безопасности в Интернете вещей

Для обеспечения безопасности отдельных смарт-устройств необходимо предусмотреть выполнение требований [a_1;a_n]. В случае, если в устройстве обеспечивается выполнение заданных требований, мы можем о говорить о «профиле безопасности» устройства – $ПХ_1$. Для обеспечения безопасности «смарт-дома» необходимо, чтобы все устройства, включенные в него, обладали «профилями безопасности», а также выполнялось дополнительное требование a_{n+1}, касающееся безопасного взаимодействия данных устройств.

Такая наследственность позволила бы оптимизировать создание систем безопасности в концепции Интернета вещей, особенно для высших уровней. Подход на основе «профилей безопасности» на практике может быть затруднен, опять же, в связи с отсутствием единых стандартов, а также закрепленных правовых или нормативных требований. Однако, 17 октября 2017 года представителями Минкомсвязи России, «Росатома», «Ростелекома», Университета ИТМО и МГУ им. М.В. Ломоносова подписан меморандум о создании Национального консорциума развития и внедрения цифровых технологий в сфере городского управления, одной из основных задач которого является реализация концепции «Умные города России». Таким

образом, ожидается, что в ближайшее время вопросы обеспечения безопасности Интернета вещей станут первостепенными и актуальность проблем значительно возрастет.

Литература:

1. Crossman M.A. and Liu H. Study of authentication with IoT [Текст]: материалы конференции/ M.A. Crossman and H. Liu; 2015 IEEE International Symposium, 2015. – P. 1-7.
2. DB Best Technologies. The Internet of Things (IoT) explained. – Redmond, 2016 [Электронный ресурс]. URL: https://www.dbbest.com/blog/the-internet-of-things/ (дата обращения 11.11.2017).
3. Preethy Wilson. Inter-Device Authentication Protocol for the Internet of Things [Текст]: диссертация/ Preethy Wilson; University of Victoria, 2017. – P. 4-10.

1.7. Построения сетей связи специального назначения на основе технологий программно-конфигурируемых сетей

Локтионов О.В.

Существующие в настоящее время региональные сети связи специального назначения (СН), как собственные, так и доверенных операторов, имеют традиционную многоуровневую и многопротокольную архитектуру, которая была разработана в 70-х – 80-х годах прошлого века. В основе ее лежит стек протоколов TCP/IP и множество других сетевых протоколов и технологий, созданных к данному моменту в рамках модели взаимосвязи открытых систем Международного союза электросвязи (OSI ISO). Этим обусловлены как несомненные достоинства с позиций апробированности, унификации и распространенности, применяемых технических и технологических решений, так и наличие ряда ограничений и недостатков, присущих принятому подходу к проектированию сетевой инфраструктуры и выбору сетевого оборудования.

Среди ограничений и недостатков можно выделить следующие.

1. Многопротокольность архитектуры сетей и используемых инфокоммуникационных технологий. Совершенствование телекоммуникационных сетей в направлении увеличения их пропускной способности, количества и качества, предоставляемых пользователям телекоммуникационных услуг в рамках существующей архитектуры, приводит к постоянному возникновению новых протоколов и технологий, число которых исчисляется сотнями. Существующие протоколы и технологии разрабатывались относительно независимо друг от друга (такая изоляция лежит в основе многоуровневых моделей сетевой архитектуры OSI и TCP/IP). Причем многие из протоколов являются проприетарными. Это увеличивает сложность сетей

и порождает проблему организации взаимодействия сетевых протоколов и технологий между собой. Настройка сетевого оборудования и администрирования сети становится трудоемкими процессами, что необоснованно ужесточает требования к квалификации специалистов и усложняет их подготовку.

2. Использование на узлах разнотипного сетевого оборудования разных производителей, в том числе иностранного производства. Это приводит к тому, что функционалы различных сетевых устройств избыточны и частично дублируют друг друга, что ведет к излишним затратам. К тому же, усложняется задача сопряжения сетевого оборудования, что зачастую преодолевается путем использования дополнительных устройств сопряжения.

3. Сложность архитектуры существующих телекоммуникационных сетей, а также использование для их построения разнородного телекоммуникационного оборудования, отличающегося реализованными в нем протоколами управления (в том числе, проприетарными), вызывают проблему автоматизированного управления сетями. Эти же факторы существенно затрудняют решение задач обеспечения стабильности процессов функционирования сетей, достижения требуемых значений сетевых характеристик и показателей качества обслуживания пользователей.

4. При существующем подходе к построению телекоммуникационных сетей МВД ограничены возможности по обеспечению гибкости и масштабируемости сетевой инфраструктуры и сетевого телекоммуникационного оборудования.

Одним из перспективных направлений развития телекоммуникационных сетей является переход к новой концепции сетевой архитектуры, получившей наименование программно-конфигурируемых сетей (ПКС) [1-3, 8].

Программно-конфигурируемая сеть – это новый подход к построению архитектуры инфокоммуникационных сетей, при котором уровень управления сетью (состоянием сетевой инфраструктуры и потоками данных в сети) и уровень передачи данных (инфраструктурный уровень) разделяются за счет переноса функций управления (выполняемых в традиционной сети маршрутизаторами и коммутаторами) на отдельное центральное устройство, называемое контроллером. Такой подход позволяет уровню управления абстрагироваться от физической сетевой инфраструктуры уровня передачи данных, используя некоторое логическое представление сети.

Основные идеи, которые закладывались в ПКС, заключаются в следующем [1-3,8]:

- разделение процессов передачи и управления данными;
- централизованное управление сетью, осуществляемое с помощью специализированного контроллера ПКС;

– унифицированный открытый интерфейс между уровнем управления и уровнем передачи данных;

– виртуализация физических ресурсов сети и сетевых функций.

Архитектура ПКС имеет три уровня (рис. 1.28):

– инфраструктурный уровень (уровень передачи данных) включает набор сетевых устройств (коммутаторов, маршрутизаторов) и каналов передачи данных;

– уровень управления, представленный центральным контроллером с сетевой операционной системой (ОС), на котором реализуются функции мониторинга текущего состояния сетевого оборудования уровня передачи данных, распределения потоков в сети и функции управления сетевыми устройствами. Уровень управления взаимодействует с уровнем передачи данных посредством унифицированного открытого интерфейса, а с приложениями – с помощью программного интерфейса (API);

– уровень сетевых приложений, в которых реализуются различные функции управления сетью: управление потоками данных в сети, управление безопасностью, мониторинг трафика, управление качеством сервиса, управления политиками и так далее.

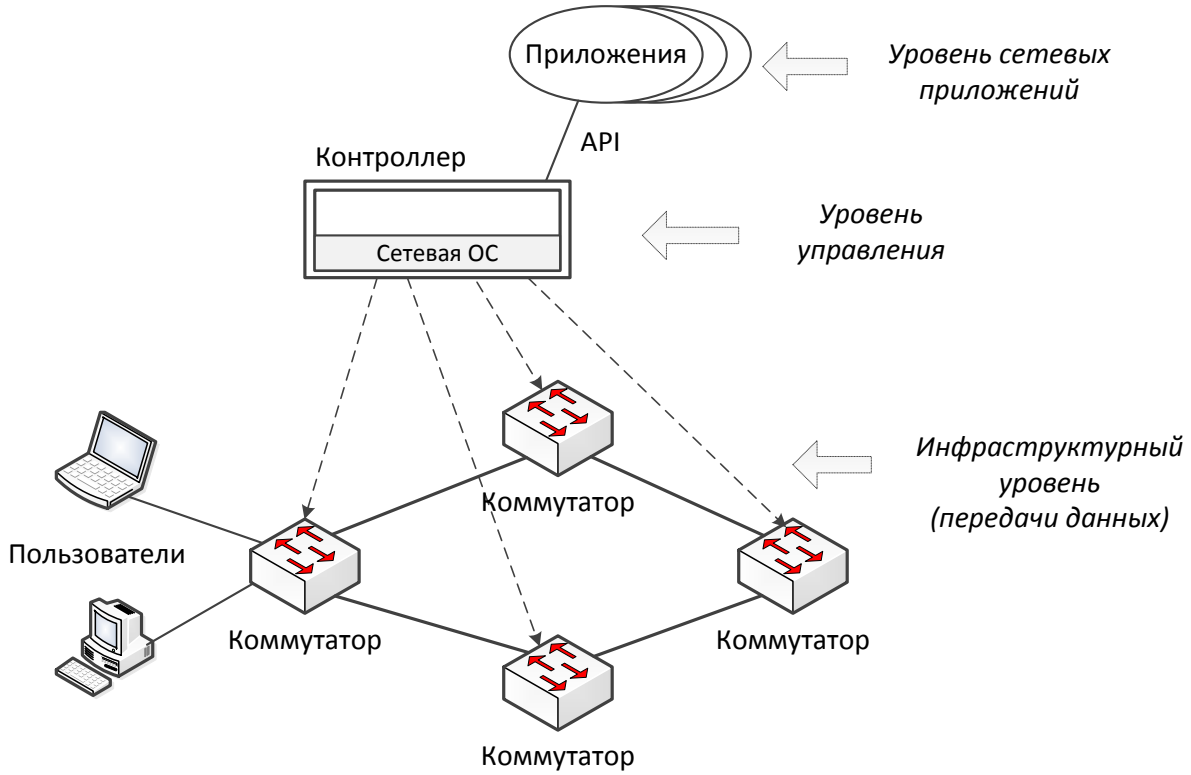


Рис. 1.28. Архитектура программно-конфигурируемой сети

В ПКС все интеллектуальные функции сетевых устройств (коммутаторов, маршрутизаторов) вынесены в контроллер, который на основе

оценки состояния сети формирует таблицы коммутации для каждого сетевого устройства и рассылает их всем узлам сети. Так формируется потоковая логическая структура сети, которая представляет собой совокупность виртуальных путей, подобно технологии многопротокольной коммутации по меткам (MPLS). При этом сетевые устройства по своему функционалу упрощаются и удешевляются. Настройка сетевого оборудования и сети в целом переносится с каждого сетевого устройства в контроллер, интеллектуальные функции которого (мониторинг сети, управление потоками и др.) выполняются программно сетевой операционной системой. Конфигурацию и настройку сети можно изменять в реальном времени, а новые приложения внедрять за гораздо более короткое время, чем в традиционной архитектуре.

Одной из наиболее перспективных и развивающихся реализаций подхода программно-конфигурируемых сетей является технология OpenFlow. Основными ее документами являются спецификации OpenFlow [6, 7], в которых описываются основные компоненты OpenFlow-сети, принципы работы и взаимодействия компонентов (протокол OpenFlow). Стандартизирующей организацией для спецификации является ONF – Open Networking Foundation.

Отдельно следует остановиться на виртуализации физических ресурсов сети и сетевых функций. Технологии виртуализации (VLAN, VPN-MPLS, VPN-IPSec, VXLAN, NVGRE, STT и др.), предполагающие решение задач логической структуризации сетей, широко используются в существующих телекоммуникационных сетях и на центрах обработки данных (ЦОД).

Виртуализация сетевых функций или NFV (network function virtualization) [3, 4], в отличие от традиционных технологий виртуализации, означает, что специализированные для выполнения отдельных функций программно-аппаратные сетевые устройства заменяются на программное обеспечение, работающее на процессорах общего назначения и устанавливаемое на комплексной платформе виртуализации сетей. NFV отличается от традиционных способов виртуализации и тем, что вместо отдельных программно-аппаратных решений для реализации каждой функции виртуализируемая сетевая функция может включать классы взаимоувязанных функций для предоставления телекоммуникационных услуг (сервисов), а также облачные инфраструктуры.

Виртуализация сетевых функций возможна во всей сети, или в тех местах, где она наиболее эффективна и экономически оправдана: в ЦОД, в сетевых узлах и подсетях пользователей.

В инфокоммуникационных сетях СН перспективным представляется использование центров обработки данных не только для выполнения функций хранения и обработки данных, но и управления сетями, начиная

с регионального уровня и выше. Применительно к управлению виртуальными сетями в ЦОД технология ПКС позволяет вынести всю логику управления сетевым оборудованием в контроллер и таким образом централизовать и автоматизировать управление сетью.

Использование технологии ПКС позволит разбивать трафик сети на различные классы (потоки) и реализовывать свою логику управления для каждого потока или группы потоков. В качестве примера использования ПКС для разделения уровня управления в виртуальных сетях можно привести средство FlowVisor (рис. 1.29). FlowVisor выделяет заданные множества потоков в отдельные срезы сети (slices), каждый из которых имеет свое логическое представление сети и логику управления [3, 4]. Срез сети определяется множеством потоков, передаваемых в данном срезе, и логическим представлением топологии сети (маршрутизаторы, коммутаторы, порты коммутаторов, соединения).

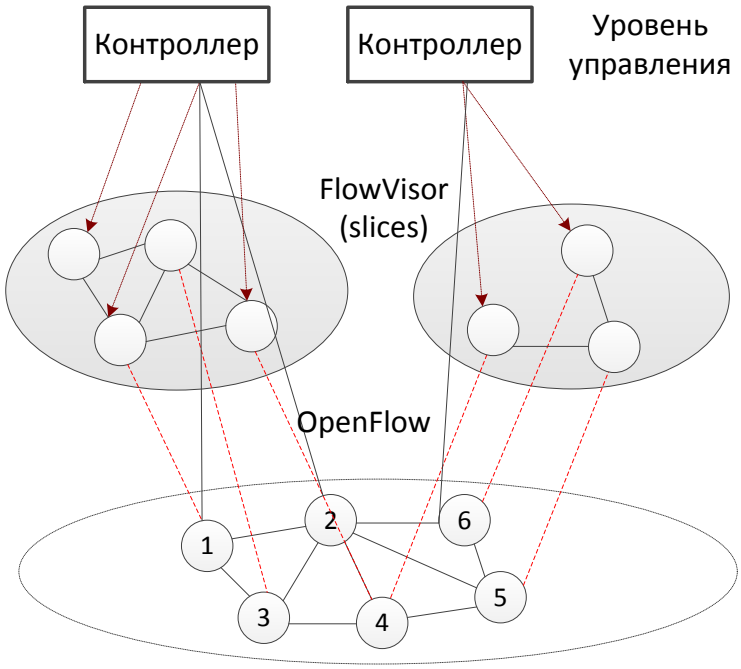


Рис. 1.29. Виртуализация сетевых функций с использованием FlowVisor

С точки зрения архитектуры ПКС, FlowVisor является прокси-сервером между сетевыми устройствами и контроллером ПКС. К одному FlowVisor может быть подключено несколько контроллеров, реализующих различную логику управления, каждый из которых управляет своим срезом сети.

FlowVisor определяет, какие потоки относятся к той или иной логической виртуальной сети и, следовательно, могут управляться соответствующим контроллером, предоставляет каждому контроллеру собственное видение логической структуры сети и обеспечивает логическое разделение

сетевых ресурсов. Таким образом, FlowVisor позволяет создавать виртуальные сети, разделяющие как уровень передачи данных, так и уровень управления.

Виртуализация сетевых функций в сетях ПКС как способа построения современной региональной сети связи СН позволит:

- повысить гибкость сетей и услуг, т.е. своевременно и легко развертывать или реконфигурировать сети, запускать новые услуги;
- повысить масштабируемость сетевых ресурсов, т.е. услуги, организованные на базе программного обеспечения, позволят более оперативно изменять объем используемых ресурсов одного и того же аппаратного обеспечения в зависимости от нагрузки;
- повысить эффективность распределения сетевых ресурсов и сбалансировать нагрузку на них;
- обеспечить не только многофункциональность, но и гибкую функциональность программно-аппаратных сетевых средств, что в свою очередь позволяет уменьшить линейку сетевого оборудования, предназначенного для выполнения различных сетевых функций. Например, отпадает необходимость создания и эксплуатации специализированных комплексов, таких как системы видеоконференц-связи, АТС, межсетевые экраны, средства трансляции адресов и т.д., так как их функции будут выполнять виртуальные машины, устанавливаемые на платформе виртуализации;
- повысить безопасность за счет изоляции потоков разных пользователей и приложений в рамках одной физической сети;
- снизить стоимость, т.к. гибкость развертывания NFV ведет к снижению расходов на управление предоставляемыми услугами и сокращает издержки, связанные с управлением всей сетью;
- сделать жизненные циклы программного и аппаратного обеспечения независимыми друг от друга.

Разработкой стандартов ПКС и изысканиями в этом направлении в настоящее время занимается большое число отраслевых объединений и международных организаций [1], в которых участвуют заинтересованные коммерческие компании-производители сетевого оборудования, операторы сетей связи, разработчики программного обеспечения (ПО): ONF, IETF, Исследовательская группа интернет-технологий (Internet Research Task Force, IRTF), Европейский институт по стандартизации в области телекоммуникаций (European Telecommunications Standards Institute, ETSI), МСЭ-Т и др.

В России исследованиями и разработкой технологий ПКС занимается Центр прикладных исследований компьютерных сетей (Сколково), который заявил о создании первого российского ПКС-контроллера Runos [5], а также ряд других организаций.

Таким образом, учитывая потенциал отечественных наработок, а также существенный положительный эффект от внедрения рассмотренного подхода можно сделать вывод о перспективности применения технологий программно-конфигурируемых сетей для построения региональных сетей связи СН.

Литература:

1. Ефимушкин В.А., Ледовских Т.В., Корабельников Д.М., Языков Д.Н. Международная стандартизация программно-конфигурируемых сетей // «ЭЛЕКТРОСВЯЗЬ» – 2014. –№ 8. – С.3-9.
2. Красотин А.А., Алексеев И.В. Программно-конфигурируемые сети как этап эволюции сетевых технологий // Моделирование и анализ информационных сетей, Т.20. – 2013. – №4. – С.110-124.
3. Смелянский Р.Л. Технологии SDN и NFV: новые возможности для телекоммуникаций. //Вестник связи. –2014. – № 1. – С.1.
4. Создание прототипа отечественной ИКС платформы управления сетевыми ресурсами и потоками с помощью сетевой операционной системы (СОС) на основе анализа и оценки существующих сетевых операционных систем для ПКС сетей и выбора одной из них для последующего развития по критериям производительности, масштабируемости надежности, безопасности. Отчет о НИР – М.: МГУ им. М.В. Ломоносова, 2013. – 252 с.
5. Центр прикладных исследований компьютерных сетей [сайт]. URL: http:// www.arccn.ru/ (дата обращения 11.11.2017).
6. ONF OpenFlow Management and Configuration Protocol (OFConfig), v.1.2., 2014 [электронный ресурс]. URL: https://www.opennetworking.org/ wp-content/uploads/2013/02/of-config-1.2.pdf (дата обращения 11.11.2017).
7. OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05) October 14, 2013 [электронный ресурс]. URL: https://www.opennetworking.org/ images/stories/downloads/sdn-resources/onf-specifications/openflow/ openflow-spec-v1.4.0.pdf (дата обращения 11.11.2017).
8. Thomas D. Nadeau, Ken Gray. SDN: Software Defined Networks. – Sebastopol: O'Reilly Media Inc., 2013. – 384 p.

1.8. Проекты SAM Cybersecurity

Семенова Т.Г., Семенова С.О.

Основные мировые вендоры программного обеспечения, такие как Microsoft, Oracle, IBM и др. разработали методики проведения проектов по управлению программными активами компаний – SAM (Software Assets

Management). Основой для методик проведения подобных проектов является серия стандартов ISO 19770 определяющих термины и понятия, описывающих функции и уровни зрелости процессов в области управления программными активами [1].

В частности, компания Майкрософт предложила программу SAM Services – это оказываемые по всему миру консалтинговые услуги, финансируемые Майкрософт. В рамках этой программы возможно проведение проектов нескольких типов: Cloud Productivity, Server Optimization, Infrastructure Optimization и др. Одним из основных и важнейших видов таких проектов является SAM Cybersecurity.

Проект SAM Cybersecurity включает в себя проведение комплексной оценки уровня защищенности корпоративной ИТ-инфраструктуры от современных киберугроз и разработку рекомендаций по устранению выявленных уязвимостей.

В ходе проекта специалисты компании-партнера[4] проводят независимое исследование IT-инфраструктуры требуемой организации, оценивает процессы обеспечения безопасности и уровень их зрелости, выявляют уязвимости в программном обеспечении внутри сети и на ее периметре путем инструментального сканирования уязвимостей рабочих станций, серверов, сетевых устройств и WEB-приложений. Затем, выполняют ранжирование уязвимостей по уровню критичности и вырабатывают рекомендации по их устранению.

Специалистами, анализируются конфигурации и программное обеспечение сетевого оборудования (межсетевые экраны, VPN, роутеры), а также установленные настройки средств защиты: антивирусных средств, групповых политик (парольная защита, аудит, контроль прав доступа, и др.), межсетевыхе экранов, шифрования, систем выявления и предотвращения вторжений (IDS/IPS), систем предотвращения утечек информации (DLP), систем сбора и анализа событий безопасности (SIEM), систем управления учетными записями (IDM) и др.

Выполняется детальное исследование отдельных аспектов обеспечения безопасности [2]:

- управление программными активами;
- защита конфиденциальной информации;
- организационно-распорядительная документация;
- безопасность баз данных;
- безопасность облачных хранилищ и др.

[4] Компании партнеры Майкрософт, участники программы MPN Microsoft Partner Network [4]

По итогам проекта разрабатываются детальные рекомендации по повышению уровня защищенности компании и устранению выявленных уязвимостей, в частности, рекомендации по переходу к следующему уровню зрелости по Critical Security Controls 20 (CSC20) [3], по закрытию уязвимостей, по интеграции средств защиты информации, по усовершенствованию процессов и процедур и другие.

Реализация проекта происходит в несколько этапов.

1. Планирование. Согласование объемов работ и приоритетов для обследования, планирование ресурсов, согласование с ключевыми лицами, подписание документов.
2. Сбор данных. Разворачивание средства сканирования и анализа защищенности, интервьюирование ответственных лиц, получение данных о серверном и сетевом оборудовании (технические характеристики), сбор информации о практиках обеспечения кибербезопасности, используемых в компании, сканирование согласованного перечня узлов локальной сети и периметра с целью обнаружения уязвимостей.
3. Анализ данных. Оценка рисков информационной безопасности и уровня зрелости процессов обеспечения кибербезопасности в компании.
4. Разработка рекомендаций. Разработка рекомендаций по повышению уровня защищенности компании с учетом организационных и технических средств, подготовка отчета.
5. Обсуждение результатов проекта. Презентация отчета и его согласование, анализ рекомендаций и последующих мероприятий, повышающих уровень защищенности.

Результатом проекта является формирования отчета, который включает в себя информацию об уровне кибербезопасности организации с соответствующими рекомендациями. Помимо этого, отчет содержит описание выявленных рисков, в том числе рисков, связанных с наличием уязвимостей в устаревшем программном обеспечении, оценку уровня зрелости процессов обеспечения безопасности, в соответствии с CSC 20 [3], а также рекомендации по повышению общего уровня защищенности IT-инфраструктуры.

Преимущества, которые получают клиенты, принимающие участие в проекте SAM Cybersecurity:

– независимая и объективная оценка уровня защищенности компании от актуальных киберугроз;

– оперативное выявление уязвимостей на рабочих станциях, серверах и сетевых устройствах;

– оценка уровня зрелости процессов обеспечения безопасности в компании;

– систематизации и упорядочивания существующих мер защиты информации;

– детальные рекомендации по снижению выявленных рисков кибербезопасности и мошенничества с использованием информационных технологий;

– понимание того, какие технологии и продукты Microsoft могут быть использованы для обеспечения защищенности компании в условиях меняющихся киберугроз;

– создание экономически эффективной инфраструктуры для обеспечения кибербезопасности.

Таким образом, проект SAM Cybersecurity является передовой методологией, позволяющей избежать множества экономических рисков, связанных с информационной безопасностью организации.

Литература:

1. ГОСТ Р ИСО/МЭК 19770-1-2014 Информационные технологии (ИТ). Менеджмент программных активов. Часть 1. Процессы и оценка соответствия по уровням. – М.: Стандартинформ, 2016. – 108с.
2. Управление программными активами для оценки кибербезопасности [электронный ресурс]. URL: https://www.microsoft.com/ru-ru/sam/cybersecurity.aspx (дата обращения: 14.10.2017).
3. CIS Critical Security Controls [электронный ресурс]. URL: https://www.sans.org/critical-security-controls. (дата обращения: 25.10.2017).
4. Microsoft Partner Network [сайт]. URL: https://partner.microsoft.com. (дата обращения: 26.10.2017).

ГЛАВА 2. МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

2.1. Вопросы практического использования российской криптографии в среде операционных систем Windows

Васильева И.Н.

Криптография является традиционным средством обеспечения конфиденциальности как хранимой, так и передаваемой информации. Однако круг задач, решаемых с использованием криптосистем, гораздо шире: контроль целостности, аутентификация сторон коммуникации, формирование общего секрета, обеспечение невозможности отказа сторон от авторства. Поэтому не удивительно, что большинство современных развитых систем, таких как операционные системы (ОС), web-серверы и СУБД, обладают встроенными криптографическими функциями.

В корпоративной среде аутентификация пользователей и устройств, работа с цифровыми сертификатами и защищенный обмен информации по сети, могут выполняться средствами базового программного обеспечения, например, ОС Windows и поддерживаемыми ею серверными службами [11]. ОС семейства Windows имеют ряд встроенных криптографических средств, поддерживающих локальное шифрование файлов (файловая шифрующая система EFS) и дисков (BitLocker), защищенную сетевую передачу информации (поддержка протоколов TLS и SSL для HTTPS, IPSec), сетевую аутентификацию по протоколу Kerberos, управление сертификатами пользователей, устройств и служб (служба Certification Authority). Некоторые из этих средств допускают настройку, предполагающую выбор криптографических алгоритмов, режимов их работы и длин ключей. Набор доступных криптоалгоритмов зависит от конкретной реализации. К сожалению, популярные криптографические функции, встроенные в ОС Windows, равно как и встроенные реализации сетевых защищенных протоколов по умолчанию не поддерживают выбор отечественных криптоалгоритмов.

Вместе с тем, криптографические методы и средства защиты информации традиционно являются объектом правовых ограничений. Так, средства криптографической защиты информации (СКЗИ), используемые в государственных информационных системах, а также для обеспечения конфиденциальность информации, доступ к которой ограничен федеральными законами РФ, должны быть сертифицированы по соответствующему классу безопасности. Это накладывает ограничения на используемые криптографические системы, а именно, СКЗИ должны реализовывать криптоалгоритмы, описанные российскими стандартами [2-5]. Использование отечественных криптографических стандартов рекомендовано и в некоторых

других случаях, например, в защищенных информационных системах финансово-кредитных учреждений РФ, для формирования квалифицированной цифровой подписи, работы удостоверяющих центров и т.д.

В настоящее время на рынке представлен широкий спектр криптографических решений, поддерживающих отечественную криптографию, – от криптопровайдеров, являющихся, по сути, низкоуровневыми библиотеками криптографических функций, отдельных утилит, надстроек и плагинов, расширяющих функции конкретных приложений или протоколов, до интегральных программно-аппаратных комплексов. Вместе с тем, применение российских СКЗИ зачастую сопряжен с рядом трудностей, вызванных:

- неразвитостью и ограниченностью пользовательского интерфейса и интерфейса администратора;
- непрозрачностью настройки;
- слабой совместимостью, а иногда и явными конфликтами, средств различных производителей между собой, а также и с базовым программным обеспечением, в частности, при его обновлении;
- ограниченной функциональностью по сравнению со встроенными механизмами защиты;
- отсутствием развитой документации по использованию и администрированию;
- слабая техническая поддержка в нестандартных ситуациях.

Стоит также добавить, что внедрение полнофункциональных программно-аппаратных комплексов криптографической защиты является достаточно дорогостоящим решением и может оказаться экономически невыгодным в условиях отсутствии необходимости обязательного выполнения требований регуляторов. Поэтому очень заманчивым было бы применение интегрированных в базовое программное обеспечение криптографических механизмов в сочетании с российскими криптоалгоритмами ГОСТ. Такой подход позволяет использовать привычную среду настройки и управления компонентов Windows с возможностью использования отечественной криптографии.

В единой среде операционной системы Windows, приложения могут обращаться к низкоуровневым реализациям криптографических функций посредством прикладного интерфейса Cryptography Next Generation (CNG) API (Crypto API в старых версиях ОС). Интерфейс CNG API позволяет разграничить прикладной уровень и уровень реализации криптографических функций (рис. 2.1), обеспечив доступ к последнему через набор стандартных функций или интерфейсов.

Поставщиками криптографии в архитектуре CNG API являются криптопровайдеры CSP, что позволяет использовать разные криптографические алгоритмы и различные реализации этих алгоритмов, включая аппаратные.

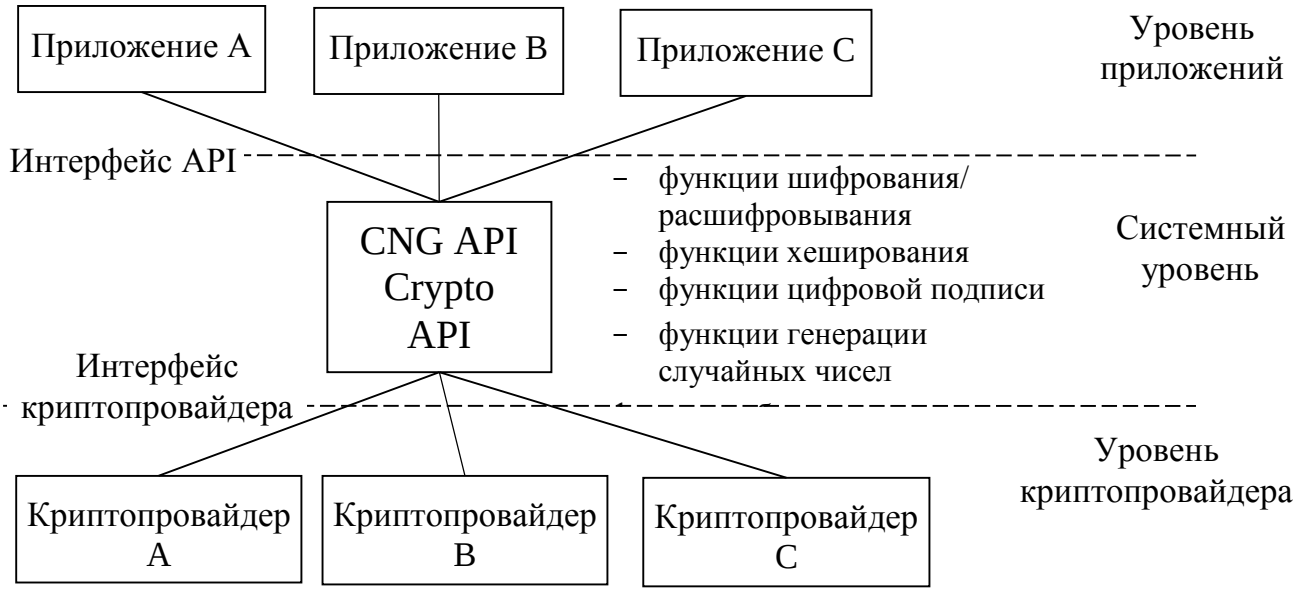


Рис. 2.1. Общая архитектура криптографических интерфейсов API

Криптопровайдер – предоставляющая специальный интерфейс и специальным образом зарегистрированная в ОС библиотека, которая позволяет расширить список поддерживаемых алгоритмов. При этом CNG API поддерживает обращение как к встроенным CSP Windows, так и к криптопровайдерам сторонних производителей. Это позволяет, установив в операционную систему сертифицированный криптопровайдер отечественных производителей (например, КриптоПро CSP, ViPNet CSP и т.п.), строить защищенные системы с поддержкой российской криптографии. Криптопровайдеры де-факто стали стандартом СКЗИ, однако несовершенство предлагаемых ОС Windows механизмов расширения вынуждает разработчиков дополнительно модифицировать высокоуровневые криптобиблиотеки и приложения MS Windows, что требует в некоторых случаях использования дополнительных утилит (например, КриптоПро IPSec, КриптоПро EFS и др).

Рассмотрим далее совместное использование службы сертификации MS Windows (Certification Authority, CA) и криптопровайдера КриптоПро CSP, как продукт одного из наиболее авторитетных отечественных производителей с сфере разработки СКЗИ (компании КриптоПро). Отметим, что альтернативой использованию CA для управления сертификатами является установка удостоверяющего центра КриптоПро УЦ. В таком случае работа с сертификатами будет осуществляться через интерфейс продуктов КриптоПро.

Служба сертификации CA является основой развертывания корпоративной инфраструктуры открытых ключей PKI и осуществляет выпуск и

управление сертификатами пользователей, компьютеров, служб и устройств в локальной сети предприятия, что позволяет на ее основе осуществлять аутентификацию, локальное шифрование и защищенный сетевой обмен данными. Поэтому важным этапом является планирование инфраструктуры PKI, что предполагает, в частности получение ответа на вопрос о необходимости внедрения в корпоративной среде:

- защищенных механизмов сетевой аутентификации на основе сертификатов, в том числе с использованием смарт-карт;
- защищенных сетевых протоколов (SSL/TLS, IPsec);
- защищенных запросов чтения и записи данных в Active Directory с помощью Secure LDAP;
- защищенной электронной почты с возможностью шифрования и/или подписания сообщений;
- подписание программного кода приложений или документов;
- управление шифрующей файловой системой EFS.

Планирование самой службы CA предполагает, исходя из анализа существующей сетевой инфраструктуры и политики безопасности компании, определение:

- иерархической структуры корневых и подчиненных центров сертификации в сети предприятия;
- криптографических алгоритмов;
- сроков действия сертификатов;
- возможности централизованной архивации и восстановления закрытых ключей,
- возможности автоматической регистрации и обновления сертификатов и т.п.

Процесс установки службы сертификации CA, поддерживающей сертификаты с российскими криптоалгоритмами, в целом следует стандартной процедуре, однако следует обратить внимание на следующие моменты:

- использование настраиваемых шаблонов сертификатов, позволяющих выбрать российские криптоалгоритмы, доступно только в версиях Enterprise/ Datacenter операционной системы Windows Server не ниже 2003;
- в системе должен быть предварительно установлен криптопровайдер КриптоПро CSP;
- должен быть выбран тип установки CA Enterprise (Предприятие);
- при создании нового ключа сертификата CA в окне настроек криптографии следует выбрать в качестве поставщика служб шифрования (CSP) отечественный криптопровайдер и включить флаг Allow administrator interaction when the private key is accessed by the CA (Разрешить взаимодействие с администратором, если ЦС обращается к закрытому ключу).

Последнее необходимо, так как отечественные криптопровайдеры используют для создания ключей биометрический генератор случайных последовательностей, что требует выполнения определенных действий со стороны пользователя.

Кроме того, установке CA может предшествовать настройка групповой политики домена, а также настройка межсетевого экрана. Например, если планируется управление сертификатами файловой шифрующей системы EFS на компьютерах домена, может быть предложена следующая процедура развертывания службы сертификации:

1. Настройка групповой политики безопасности, запрещающей использование файловой шифрующей системы EFS на компьютерах домена (ветвь Computer Configuration/ Policies/ Widows Settings/ Security Settings/ Public Key Policies/ EFS File System – Конфигурация компьютера/ Политики/ Конфигурация Windows/ Параметры безопасности/ Политики открытого ключа/ Шифрующая файловая система EFS).

2. Установка криптопровайдера КриптоПро CSP и утилиты КриптоПро EFS на сервере и клиентских компьютерах.

Здесь стоит отметить, что не все компоненты Windows позволяют осуществлять выбор криптопровайдера с помощью шаблонов безопасности. В таких случаях должны быть установлены дополнительные утилиты КриптоПро, использующиеся совместно с криптопровайдером. Такие утилиты фактически представляют собой надстройку над механизмами реализации этого компонента (например, EFS) в операционной системе.

3. Установка на сервере центра сертификации CA с выбором российских криптографических алгоритмов. При установке кроме службы центра сертификации Certification Authority может быть также установлена служба Certification Authority Web Enrollment (Служба регистрации в центре сертификации через Интернет).

Использование службы Web Enrollment может быть полезно при работе с сертификатами вне домена (когда работа с консолью mmc недоступна). Вместе с тем, эта служба имеет ряд ограничений, в частности, через web-интерфейс сложно получить сертификаты со значениями параметров, отличными от заданных в шаблоне. Кроме того, не поддерживается работа с шаблонами V3 (Windows Server 2008 Enterprise).

4. Настройка шаблонов сертификатов. Перед выдачей сертификатов следует настроить шаблоны, на основе которых они будут создаваться. Шаблоны сертификатов настраиваются в оснастке центра сертификации. Необходимо выбрать те шаблоны, которые соответствуют планируемым сервисам безопасности, например, для использования шифрования EFS следует создать дубликат и настроить параметры шаблонов EFS Basic (Базовое шифрование EFS) и EFS Recovery Agent (Агент восстановления EFS).

Для использования российских криптоалгоритмов следует создать копии указанных шаблонов, выбрав legacy-тип, то есть шаблон V2 (Windows Server 2003 Enterprise).

Следует отметить, что CNG позволяет разработчикам запрашивать алгоритмы, не указывая поставщиков алгоритмов. Шаблоны V3 содержат предопределенные перечни алгоритмов (цифровой подписи, шифрования с открытым ключом, хэширования) и не предоставляют возможности выбора криптопровайдера. Поэтому для целей использования отечественной криптографии шаблоны V3 не подходят. Настройка шаблонов V2 для этих целей так же имеет ряд особенностей:

- создание экспортируемых ключей не поддерживается – следует отключить флаг Allow private key to be exported (Разрешить экспортировать закрытый ключ);
- поскольку отечественный стандарт ГОСТ Р 34.10–2012 определяет цифровую подпись на эллиптических кривых с ключами размером 512 или 1024 бит, в шаблоне требуется установить минимально возможный размер ключа (512 бит);

Правильность указания этих параметров определяет возможность выбора отечественных CSP в качестве поставщиков криптоалгоритмов, в противном случае они не будут отображены в списке доступных криптопровайдеров.

Для каждого шаблона следует настроить политику выдачи, указав группы или пользователи, имеющие доступ к шаблону. Для запроса и получения сертификатов достаточно дать права Enroll (Заявка) и, возможно, AutoEnroll (Автоматическая подача заявок).

Право на автоматическую подачу заявок является дополнительным к праву заявки, то есть не будет действовать без него. Следует также отметить, что автоматическая регистрация для сертификатов на основе настраиваемых шаблонов сертификатов с поддержкой российской криптографии будет действовать только при соблюдении следующих условий:

- для шаблона указана необходимость запроса пользователя при создании сертификата (установлен флаг Prompt the user during enrollment – Запрашивать пользователя во время регистрации);
- в домене включена групповая политика автоматической регистрации сертификатов (ветвь User configuration/ Policies/ Widows Settings/ Security Settings/ Public Key Policies/ Certificate Services Client - Auto-Enrollment – Конфигурация пользователя/ Политики/ Конфигурация Windows/ Параметры безопасности/ Политики открытого ключа/ Клиент служб сертификации: автоматическая регистрация, дополнительно следует установить флаг Update certificates that use certificate templates – Обновлять сертификаты, использующие шаблоны сертификатов).

После настройки следует указать, что созданные шаблоны будут служить для выпуска сертификатов CA (New/ Certificate Template to Issue – Создать/ Выдаваемый шаблон сертификатов), исключив исходные шаблоны из списка выдаваемых.

5. Получить сертификат агента восстановления EFS, используя оснастку Сертификаты. Роль агента восстановления EFS по умолчанию доступна для администраторов, однако рекомендуется для этих целей создать отдельную учетную запись. Запрос сертификата с помощью оснастки Сертификаты позволяет изменить значения некоторых параметров шаблона, в частности осуществить выбор:

- криптопровайдера из списка доступных;
- алгоритма шифрования;
- длины ключа.

Так же существует возможность запросить сертификат с экспортируемым закрытым ключом, даже если это свойство отключено в шаблоне сертификата.

Закрытый ключ сертификата агента восстановления EFS рекомендуется экспортировать на внешний носитель, чтобы обеспечить возможность его восстановления из файла при необходимости. Файлы-контейнеры закрытого ключа рекомендуется хранить на защищенных носителях, таких как смарт-карта или USB-токен. В этом случае сертификат может быть перенесен на защищенный носитель сразу же в процессе его создания. При одновременном удалении закрытого ключа сертификата из системы агент восстановления становится полностью независимым от своего профиля.

При использовании защищенных носителей (смарт-карты, USB-токена) предварительно должен быть настроен шаблон типа Smart Card Logon (Вход со смарт-картой) или Smart Card User (Пользователь со смарт-картой) и получен соответствующий сертификат.

6. Настроить групповую политику файловой шифрующей системы EFS в домене, указав полученный сертификат в качестве сертификата агента восстановления данных и разрешив шифрование EFS, сняв флаг Allow EFS to generate self-signed certificates when the certification authority is not available (Разрешить EFS создавать самозаверяющиеся сертификаты, если центр сертификации недоступен).

Теперь агент восстановления сможет получить доступ ко всем файлам, зашифрованным с использованием сертификатов, которые будут выданы позднее. При этом пользователи домена смогут использовать шифрование только на основе сертификатов, выданных CA.

7. При необходимости настроить службу Certification Authority Web Enrollment, добавив в настройках сайта по умолчанию web-сервера IIS, расположенного на сервере с CA, поддержку протокола HTTPS. В качестве

сертификата можно использовать сертификат компьютера сервера или сертификат web-сервера, предварительно настроив соответствующий шаблон.

8. До начала использования шифрующей файловой системы EFS получить сертификаты шифрования от имени пользователей. Сертификаты могут быть получены через web-интерфейс (без поддержки экспорта закрытого ключа) либо с помощью оснастки Сертификаты. Если включена поддержка автоматической регистрации, при первом входе в систему пользователю будет предложено сформировать запрос на получение сертификата (запрос оснастки Сертификаты).

Подобная процедура позволяет использовать стандартный интерфейс ОС Windows для шифрования файлов и папок, предоставления доступа ограниченного числа пользователей к зашифрованному файлу (путем выбора их сертификатов в свойствах зашифрованного файла), а также восстанавливать файлы от лица агента восстановления EFS в случае утраты сертификата шифрования. Шифрование в сетевых папках поддерживается ограниченно. Кроме того, при отчуждении сертификата из системы (например, экспорта на защищенный внешний носитель при одновременном удалении из системы) доступ к зашифрованным данным становится невозможным даже в том случае, если был получен доступ к профилю пользователя. Поэтому даже без использования смарт-карт или usb-токенов сотрудник, например, уезжая в отпуск, может удалить закрытый ключ из системы, сделав невозможным просмотр своих данных. После обратного импорта сертификата с закрытым ключом в систему и установки из контейнера закрытого ключа через интерфейс КриптоПро CSP доступ к зашифрованным файлам восстанавливается.

Аналогично описанной процедуре могут быть настроены шаблоны и получены сертификаты для использования в защищенном сетевом обмене по протоколу IPSec, аутентификации Kerberos и т.д. Отметим также, что указанные сертификаты могут использоваться и для подписания документов и сообщений в приложениях MS Office при установке надстройки КриптоПро Office Signature. Вместе с тем следует отметить, что сертификаты КриптоПро имеют ряд ограничений – так, не поддерживается централизованная архивация и восстановление закрытых ключей (в CA для этих целей предусмотрена роль агента восстановления ключей), шаблоны V3, а на шаблоны V2 также накладывается ряд ограничений, которые были рассмотрены выше.

Интересно, что выбор стороннего CSP и отечественных алгоритмов шифрования потенциально возможен даже для парольной защиты документов Microsoft Office, поскольку процедура шифрования документа не накладывает ограничений на используемый блочный симметричный шифр или хэш-функцию [14]. Ранние версии MS Office имели слабую

криптографию, реализуя такие алгоритмы, как побитовый XOR с обфускацией, RC4 с 40-битным ключом, а длина пароля была ограничена 16 символами. В большинстве случаев парольная защита документов могла быть легко взломана методом грубой силы. Начиная с версии MS Office 2010 используются алгоритмы, поставляемые на уровне ОС криптопровайдерами CSP, а схема генерации ключа значительно усилена для снижения эффективности подбора паролей (рис. 2.2). Кроме того для зашифрованных данных производится проверка целостности с использованием конструкции HMAC.

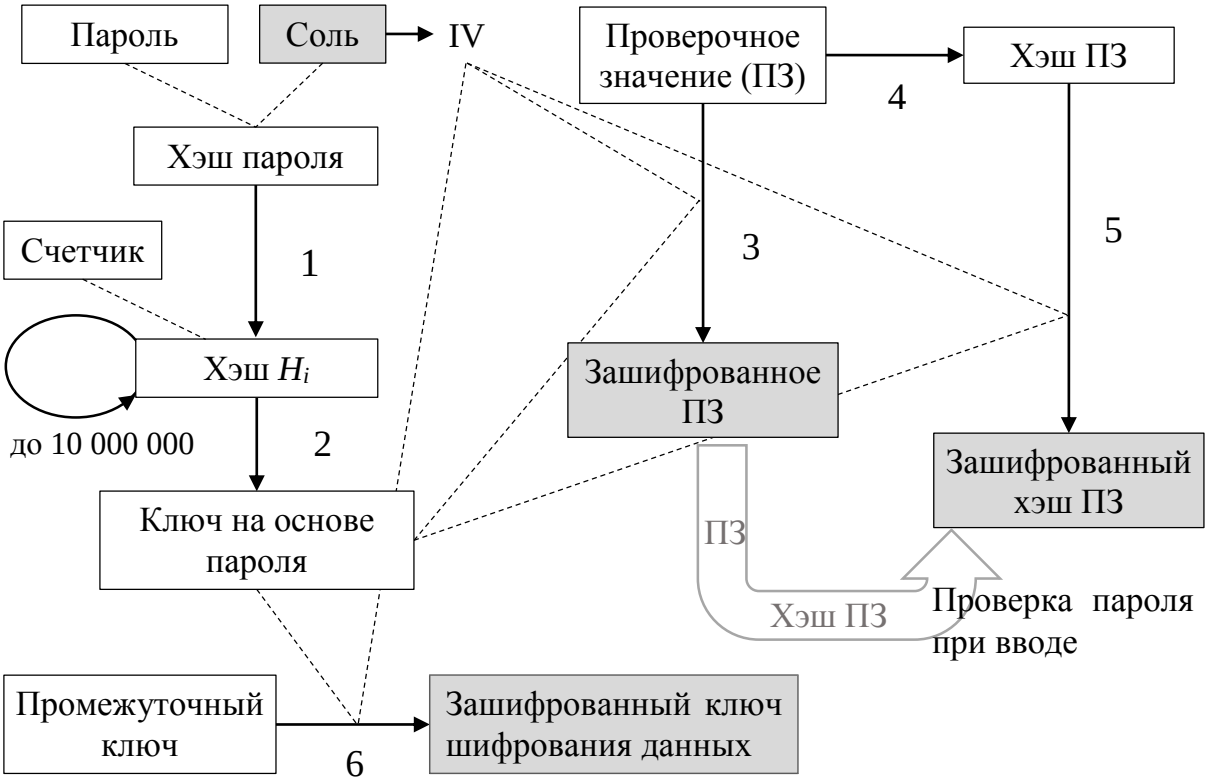


Рис. 2.2. Генерация ключевой информации в документах MS Office

Пароль пользователя дополняется случайной последовательностью (соль) и хэшируется, затем процедура хэширования итерационно повторяется (шаг 1), причем входные данные каждый раз дополняются нарастающим значением счетчика. Финальная итерация использует в качестве дополняющего значение специального вида. Ключ на основе пароля получается из финального значения хэша путем обрезки последнего, или напротив, дополнения последовательностью фиксированного вида (в зависимости от используемой хэш-функции и требуемой длины ключа) – шаг 2. Однако полученный ключ не используется для шифрования данных. Его назначение – шифрование информации, сохраняемой вместе с

документом, а именно – значения, используемого для проверки правильности введенного пользователем пароля и его хэша, а также случайного ключа шифрования данных, который в документах Microsoft назван промежуточным ключом. По умолчанию для шифрования используется режим сцепления блоков CBC с вектором инициализации, получаемым на основе значения соли. Значения, сохраняемые вместе с документом, выделены на рис. 2.2 заливкой цветом.

Для проверки правильности пароля, введенного пользователем для доступа к защищенному документу, используется генерируемое случайным образом проверочное значение. Оно шифруется и сохраняется вместе с документом (шаг 3). Затем вычисляется хэш проверочного значения (шаг 4), который также шифруется (шаг 5) и сохраняется. Теперь после ввода пароля пользователем может быть вычислен ключ, расшифровано значение проверочного значения, вычислен его хэш, последний должен быть зашифрован и сравнен с сохраненным зашифрованным значением хэша. Совпадение значений подтверждает правильность пароля.

По умолчанию приложения MS Office используют симметричный блочный шифр AES с 128-битовым ключом и хэш-функция SHA-1. Поскольку SHA-1 генерирует хэш-код размером 160 бит, а алгоритм AES производит шифрование блоками по 128 бит, на шаге 5 приходится производить шифрование двух блоков, второй из которых является неполным и дополняется до полной длины нулевыми битами. Тогда при попытке определения ключа с помощью операции, обратной шагу 5, нарушитель получит критерий проверки правильности подбора ключа – наличие заданного количества нулей на конце расшифрованного значения [10]. Даже с учетом практической невозможности полного перебора ключей размером 128 бит, данный факт внушает определенные опасения. Поэтому можно сформулировать следующие требования к алгоритмам, используемым для парольной защиты документов MS Office:

- размер выхода хэш-функции должен быть не короче длины ключа симметричного шифра;
- размер выхода хэш-функции должен быть кратен размеру блока симметричного шифра.

Применительно к алгоритмам, используемым в MS Office по умолчанию, достаточно заменить алгоритм хэширования SHA-1 на хэш-функцию SHA-256 или SHA-512, тем более что SHA-1 в настоящее время признан небезопасным [12]. Следует отметить, что российские криптоалгоритмы полностью удовлетворяют сформулированным требованиям.

Настройка криптографии MS Office осуществляется с помощью групповой или локальной политики и загружаемых с сайта производителя шаблонов безопасности. После копирования файлов шаблонов в стандартное

расположение на компьютере параметры безопасности автоматически считываются редактором групповой политики. Для MS Office возможен выбор криптопровайдера с указанием используемого симметричного шифра и длины ключа (политика Тип шифрования для защищенных паролем файлов Office Open XML). Указанная политика распространяется на шифрование защищенных паролем документов MS Excel, PowerPoint и Word при условии использования пользовательской COM-надстройки для шифрования. COM-надстройки – механизм, позволяющий разработчику расширить функциональные возможности приложений Office для решения пользовательских задач. COM-надстройка представляет собой элемент ActiveX DLL и может быть использована после установки, регистрации в реестре Windows и активации в окне приложения. Однако подобные надстройки ведущими отечественными разработчиками СКЗИ не предоставляются.

Кроме того, для приложений MS Access, Excel, OneNote, PowerPoint, Project и Word определены собственные политики, связанные с парольной защитой. Эти политики позволяют задавать такие параметры, как:

- используемый симметричный блочный шифр (Задать алгоритм шифрования CNG);
- режим работы блочного шифра (Настройка режима цепочки шифрования CNG);
- длину ключа блочного шифра (Задать длину ключа шифрования CNG);
- используемую хэш-функцию (Задать алгоритм хэширования CNG),

а также значения некоторых параметров процедуры генерации ключа (размер соли, число итераций вычисления хэша пароля).

Перечисленные политики конкретных приложений предполагают указание имен алгоритмов без выбора криптопровайдера, что создает определенные сложности в плане использования отечественной криптографии.

Политики цифровой подписи документов MS Office не предполагают выбор криптографических алгоритмов, поскольку последние определяются имеющимися сертификатами пользователя. Для обеспечения возможности использования сертификатов с российскими криптоалгоритмами для подписывания документов MS Office компания КритоПро предлагает установить соответствующую надстройку (КриптоПро Office Signature).

Следует отметить, что само по себе использование сертифицированного поставщика криптографических функций или эталонных реализаций криптографических стандартов [8] при самостоятельной реализации приложения не гарантирует требуемого уровня защищенности. Это связано необходимостью решения ряда проблемных вопросов, и прежде всего, обеспечения корректности использования ключей, безопасности

управления ключевой информацией и взаимодействия со средой функционирования СКЗИ. Так, например, приведенная выше схема парольной защиты компании Microsoft существенно отличается от рекомендаций как американского NIST [15], так и отечественного технического комитета по стандартизации «Криптографическая защита информации» (ТК 26) [6, 9], схема генерации ключа в которых предусматривает использование не простого хэширования, а конструкции HMAC.

Множество вопросов связано и с реализацией генерации случайных значений. Например, периодически появляются сообщения о слабости шифрования документов MS Office, связанные с использованием одинаковых ключей (в частности, подобная уязвимость была обнаружена как для старых [1], так и для относительно новых [13] версий MS Excel). Встроенные генераторы «случайных» чисел, доступные в большинстве высокоуровневых языков программирования, как правило, не являются криптографически сильными. В случае же использования специальных криптографических библиотек источником проблем может быть отсутствие или не полнота документации, а также невозможность управления отдельными параметрами. Ориентиром в этом направлении являются рекомендации по стандартизации группы «Информационная технология. Криптографическая защита информации» и методические документы ТК 26, в частности проект рекомендаций по стандартизации [7].

Следует также иметь в виду, что согласно федеральному закону «О лицензировании отдельных видов деятельности» от 04.05.2011 № 99-ФЗ, разработка, производство, распространение (и др.) СКЗИ и информационных систем, защищенных с помощью СКЗИ, подлежит обязательному лицензированию.

Литература:

1. В системе шифрования Office обнаружена ошибка / Роберт Лемос (Robert Lemos), CNET News.com, 21.01.2005 электронный ресурс]. URL: http://www.astera.ru/news/?id=20957 (дата обращения: 11.11.2017).
2. ГОСТ Р34.10–2012 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – М.: Стандартинформ, 2012. – 29 с.
3. ГОСТ Р 34.11–2012 Информационная технология. Криптографическая защита информации. Функция хэширования – М.: Стандартинформ, 2012. – 34 с.
4. ГОСТ Р 34.12–2015 Информационная технология. Криптографическая защита информации. Блочные шифры. – М.: Стандартинформ, 2015. – 21 с.

5. ГОСТ Р 34.13–2015 Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров. – М.: Стандартинформ, 2015. – 38 с.
6. Парольная защита с использованием алгоритмов ГОСТ. Методические рекомендации технического комитета по стандартизации «Криптографическая защита информации» (ТК 26), 27.11.2012 [электронный ресурс]. URL: https://www.tc26.ru/methods/containers_v1/Addition_to_PKCS5_v1_0.pdf (дата обращения: 11.11.2017).
7. Принципы разработки и модернизации шифровальных (криптографических) средств защиты информации [электронный ресурс]. URL: http://www.tc26.ru/standard/gost/ (дата обращения: 11.11.2017).
8. Программная реализация криптографического преобразования базовых блочных шифров, определенных стандартом «Информационная технология. Криптографическая защита информации. Блочные шифры» и режимов их работы [электронный ресурс]. URL: http://www.tc26.ru/standard/gost/PR_GOSTR_bch_v9.zip (дата обращения: 11.11.2017).
9. Рекомендации по стандартизации Р 50.1.111–2016. Информационная технология. Криптографическая защита информации. Парольная защита ключевой информации. – М.: Стандартинформ, 2016. – 15 с.
10. Старые недоработки в MS Office на новый лад/ Pavel Semjanov, 03.12.2009 // Все о паролях и практической криптографии [электронный ресурс]. URL: http://www.password-crackers.ru/blog/?p=87 (дата обращения: 11.11.2017).
11. Чернокнижный Г.М. Вычислительные сети. Контроль безопасности в компьютерных сетях: учебное пособие – СПб.: Изд-во СПбГЭУ, 2016. – 97 с.
12. Эксперты осуществили первую успешную атаку поиска коллизий хеш-функций SHA-1, 25.02.2017 [электронный ресурс]. URL: https://www.aktiv-company.ru/press-center/publication/2017-02-25.html (дата обращения: 11.11.2017).
13. Mitsunari Shigeo, Yoshinari Takesako Backdoors with the MS Office file encryption master key and a proposal for a reliable file format, 28.10.2015 [электронный ресурс]. URL: https://www.slideshare.net/codeblue_jp/ backdoors-with-the-ms-office-file-encryption-master-key-and-a-proposal-for-a-reliable-file-format-by-mitsunari-shigeo-yoshinari-takesako (дата обращения: 11.11.2017).
14. [MS-OFFCRYPTO]: Office Document Cryptography Structure [электронный ресурс]. URL: https://msdn.microsoft.com/ru-ru/library/cc313071.aspx (дата обращения: 11.11.2017).

15. NIST Special Publication 800-132 Recommendation for Password-Based Key Derivation. Part 1: Storage Applications, December 2010 [электронный ресурс]. URL: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf (дата обращения: 11.11.2017).

2.2. Анализ и разработка системы обнаружения вторжений

Чернокнижный Г.М., Ишанханов С.Р.

Системы обнаружения вторжений (СОВ) или Intrusion Detection System (IDS) и системы предотвращения вторжений Intrusion Prevention System (IPS) предназначены, соответственно, для обнаружения и предотвращения потенциальной или реальной угрозы несанкционированного доступа в информационную систему (ИС).

Развиваются эти системы довольно давно (модель системы обнаружения вторжения Дороти Деннинг опубликовала в 1987 году [2]) и базируются на методах сигнатурного анализа и методах обнаружения аномалий (эвристических правилах).

Сигнатурный анализ в СОВ работает с известными сценариями атаки. Попытка реализации таких сценариев злоумышленником обнаруживается анализаторами трафика или анализом логов входных событий.

Обнаружение аномального поведения системы предполагает, что в обычном режиме ИС имеет некоторый «нормальный» профиль, соответствующий регулярному протеканию информационного процесса. Отклонение от нормального профиля может являться косвенным признаком атаки. Важной особенностью этого типа СОВ является необходимость в наличии механизма адаптации профиля ИС к изменению внешней ситуации. Необходимо разработать адаптивные алгоритмы, при помощи которых будет автоматически (или с вмешательством эксперта) составляться профиль реальной работающей системы. Это нужно для того, чтобы «научить» СОВ различать штатный режим работы ИС при изменении ситуации. Преимуществом использования рассматриваемого подхода является теоретическая возможность обнаружения новых, не описанных ранее атак.

Использование систем предотвращения вторжения преследует несколько целей:

- обнаружение и предотвращение вторжений;
- получение информации о вторжениях и выявление причин проникновения;
- анализ уязвимостей, которые привели к проникновению;
- документирование существующих угроз;
- получение сведений о злоумышленнике;
- обеспечение контроля качества администрирования.

Типовая СОВ включает следующие компоненты:

- подсистема сбора событий;
- подсистема анализа собранных событий;
- база накопленных данных;
- базы данных уязвимостей;
- консоль управления.

Схема сетевой СОВ показана на рис. 2.3.

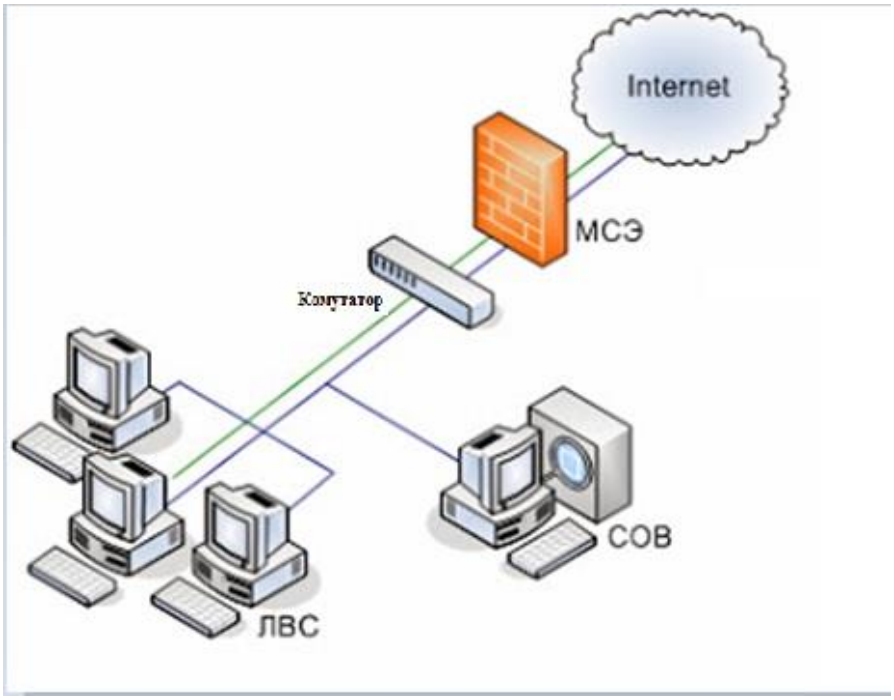


Рис. 2.3. Схема сетевой СОВ

С учетом предполагаемого использования системы – малые и средние предприятия, в работе была поставлена цель – создать систему обнаружения вторжений в бюджетном варианте на базе свободно распространяемых программных модулей.

Основой любого программного комплекса является системное программное обеспечение (ПО). Естественным выбором операционной системы (ОС), как платформы для СОВ, стало семейство ОС Linux. Конкретно была использована ОС Ubuntu (последнее ядро Zetsy Zapus). Помимо добавления стандартного набора пакетов, который имеется в любом дистрибутиве Linux, в систему был добавлен набор пакетов, позволивший превратить систему в мощный инструмент анализа трафика и поиска уязвимостей. Важнейшими из дополнительного набора являются модули IDS/IPS Bro (Bro-core, Bro-script, Bro-engine, Bro-syst, Libpcap). Система Bro разрабатывается в Калифорнийском университете в Беркли и в настоящее время используется в проектах многих серьезных американских компаний.

Bro представляет собой систему для создания сетевой IDS/IPS и имеет многоуровневую модульную структуру [1]:

- механизм захвата пакетов, который использует для данных целей libpcap (библиотеку с открытым исходным кодом), что позволяет Bro не зависеть от платформы и от нижележащего сетевого уровня. В этом функционале Bro может заменять известные снифферы, например, Wireshark, выделяя и анализируя только необходимый трафик;
- механизм событий (EventEngine) преобразует пришедшие последовательности пакетов в первичные события. События эти отражают базовые сведения о сетевой активности. Например, каждый HTTP-запрос порождает соответствующее событие, которое описывает адрес, порт, запрашиваемый URL и версию протокола HTTP. Этот механизм, однако, не принимает никаких решений относительно оценки события — то есть на данном уровне неизвестно, вредоносное оно или нет;
- верхний уровень, интерпретатор скриптов (PolicyScriptInterpreter): каждая реакция на какое-либо событие регистрируется его обработчиком, соответствующим определенному скрипту. События ставятся в очередь FIFO. Скрипты же определяют действия, использующиеся для обнаружения вредоносного трафика, а также политику, применяемую при его обнаружении, и пишутся на собственном скриптовом языке Bro. Отметим, что в исходном модуле в виде библиотеки имеется набор готовых скриптов. Пример некоторых скриптов приведен на рис. 2.4.

```
# Скрипт, логирующий остальные загружаемые скрипты
@load misc/loaded-scripts
# Скрипты, задающие стандартные параметры некоторых тонких настроек
@load tuning/defaults
# Скрипт обнаружения сканирования
@load misc/scan
# Логируем некоторую информацию о веб-приложениях, используемых в данной сети
@load misc/app-stats
# <...>
# Обнаруживает софт, применяемый для различных протоколов
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
@load protocols/http/software
# <...>
# Обнаруживает попытки SQL-инъекций
@load protocols/http/detect-sqli
# Считываем хеши всех файлов, которые проходят через Bro, и отправляем их в сервис Detect-MHR (Malware Hash Registry)
@load framework/files/hash-all-files
@load framework/files/detect-MHR
# <...>
```

Рис. 2.4. Пример готовых скриптов системы Bro

Кроме указанных модулей, в систему включен ряд важных компонент, в частности: Open SSl.lib, Bin8 Library, Libz, Libmagic, Python, Lib Geo IP, Sendmail и др., которые повышают функциональность, гибкость и защищенность системы.

Помимо перечисленных системных модулей в ОС добавлены пакеты, обеспечивающие сервисы и функциональность. В качестве рабочего стола использована графическая оболочка Gnome 3, представленная набором модулей, которые можно использовать при создании нужного профиля.

Защищенность самой системы повышена выполненными настройками:

- надежный пароль пользователя root с регулярным обновлением;
- установка пароля для редактирования загрузчика;
- ролевая модель доступа к приложениям;
- криптографическая верификация пакетов;
- использование ключей для соединения по протоколу SSH;
- шифрование диска с помощью алгоритма Linux Unified Key Setup-on-disk-format (LUKS);
- автоматическое обновление настроек безопасности с серверов Ubuntu.

После конфигурирования было проведено тестирование системы и ее реакция на различные виды атак. При этом задействовались встроенные возможности системы на базе готовых скриптов. В частности, с помощью программы nmap проводилось стелс-сканирование (Stealth Scan, скрытое сканирование) портов созданной системы. СОВ использовала сигнатурный анализ данных потока автоматически, без вмешательства администратора. Затем программа переходила в режим анализа активности портов и выводила данные в журнал, в котором отмечалось количество попыток сканирования портов и выводилось системное сообщение «notice». Атаки DDOS, IP-спуффинг, Bruteforce и еще ряд атак проводились с помощью виртуальной машины Kali Linux. Система использовала для обнаружения атак скрипты Bro.

Таким образом, при бета-тестировании разрабатываемая СОВ показала весьма неплохие результаты. Однако, как отмечалось выше, необходимо дополнить систему алгоритмами адаптации профиля информационной системы к изменению внешней ситуации. Поэтому в развитие СОВ ставится задача создания алгоритма самообучения системы.

Были проанализированы характеристики известных СОВ, предлагаемых зарубежными вендорами: Cisko, Symantec, IBM Internet Security Systems, McAfee, Check Point IPS и др. а также отечественными: Entensys Corporation, Infotecs, Код Безопасности, Компания РНТ и др. Большинство

из разработчиков используют оба метода обнаружения атак и дополняют основной функционал новыми возможностями, например, межсетевым экраном, VPN, встроенным антивирусом. Следует отметить, что все больше СОВ имеют программно-аппаратные решения.

На основании анализа были отмечены два главных направления развития современных СОВ:

- создание и развитие методов и средств обнаружения атак;
- совершенствование способов реагирования на выявленные атаки.

Рассмотрим первое направление. Если сигнатурный анализ требует только постоянного поддержания в актуальном состоянии базы сценариев атак и совершенствования соответствующих движков, то в развитии эвристических методов наблюдаются различные подходы к разработке адаптивных алгоритмов обучения СОВ путем создания элементов искусственного интеллекта. К таким исследованиям относятся:

- применение нейронных сетей;
- методы построения иммунных систем;
- применение генетических алгоритмов;
- Data Mining – интеллектуальный анализ данных (обнаружения в базах данных нетривиальных и практически полезных закономерностей).

Перечисленные исследования привносят в создание СОВ принципы, заимствованные из биологических систем. Наиболее обучаемым интеллектуальным средством для решения задач классификации являются нейронные сети [6]. Первое упоминание о предложении использовать нейронные сети в СОВ датируется 1993 годом [3]. Преимуществами нейросетевых вычислений являются адаптивность, информационная защищенность, параллелизм, способность выделения скрытых в информации знаний.

Было доказано, что любую функцию можно представить в виде многослойной нейронной сети из формальных нейронов с нелинейной функцией активации [5]. Каждый слой такой сети (персептрона) состоит из нейронов, которые взаимосвязаны со всеми нейронами в последующем слое. Нейронная сеть, использующая большое количество скрытых слоев называется глубокой нейронной сетью. Анализ доступных источников показал, что наилучшие результаты в обучении дает алгоритм обучения методом обратной ошибки, который вычисляет ошибки каждого нейрона, ошибку выходного слоя, а также корректирует веса нейронов. Входной слой в данной системе имеет восемь нейронов, выходной слой имеет два нейрона, соответствующие нормальному функционированию и работе при атаке злоумышленника (рис. 2.5).

Персептрон имеет три скрытых слоя: первый анализирует трафик на предмет аномалий, второй идентифицирует атаку по паттернам (шаблонам поведения), третий определяет степень участия инсайдеров в аномальном

поведении системы. Здесь используются библиотеки на языке Python, в которых реализовано большое количество алгоритмов машинного обучения: NumPy, Keras, Scikit-learn.

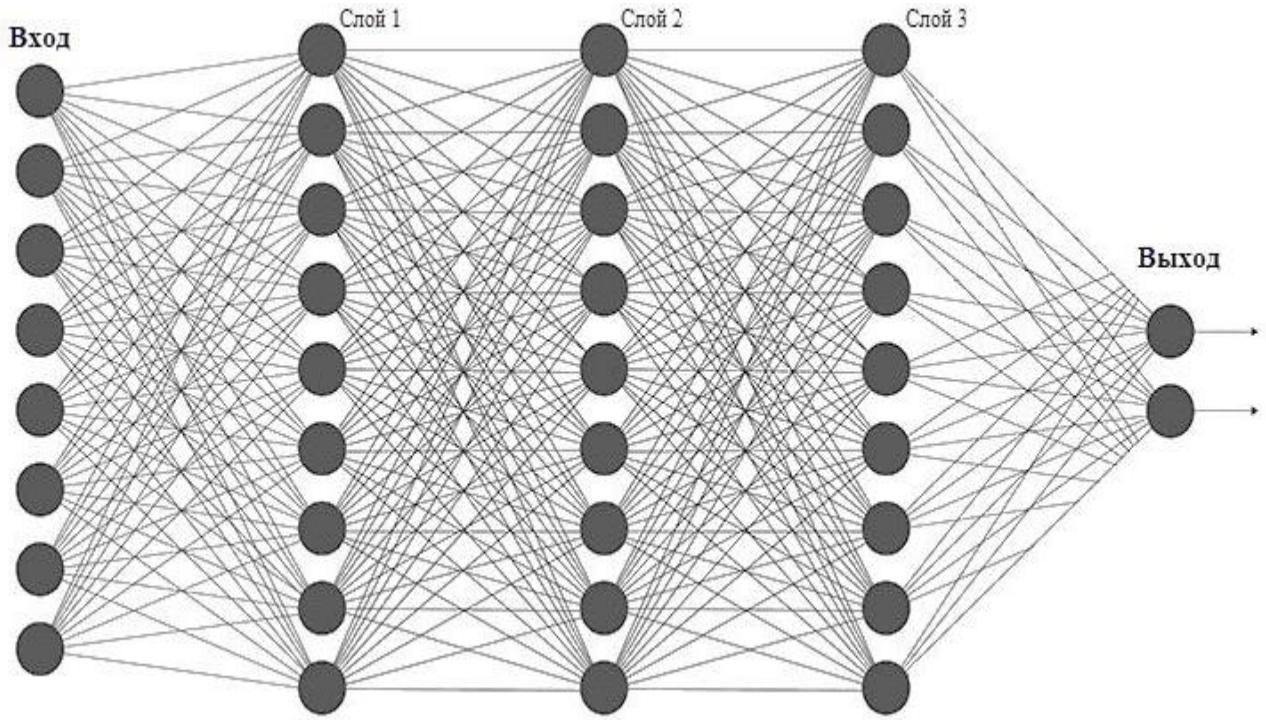


Рис. 2.5. Вид персептрона для разрабатываемой СОВ

Однако особенность функционирования рассматриваемых систем состоит в том, что они должны работать в условиях постоянно меняющейся внешней ситуации, т.е. неопределенности. Поэтому для достижения цели адаптации и обучения СОВ представляется наиболее подходящим вариантом применение сочетания достоинств нейронных сетей и опыта человека-эксперта, специалиста по информационной безопасности. Это, в свою очередь, предполагает использование аппарата нечеткой логики. Подобные модели рассматривались в работах зарубежных и отечественных авторов, в частности, в [7].

Механизм нечеткого логического вывода позволяет использовать для предварительного обучения опыт экспертов, сформулированный в виде системы нечетких предикатных правил. Это даст возможность автоматически создавать новые правила при появлении новых атак. В качестве входной информации используются описываемые экспертами атаки, формализованные в виде системы нечетких правил (условий). Применительно к анализу сетевых пакетов эти правила могут включать определенные значения отдельных полей заголовка пакета (IP-адрес и порт источника или получателя, установленные флаги, размер пакета и т. д.). При анализе журналов

регистрации событий правила могут ограничивать время регистрации пользователя в системе, количество попыток неправильного ввода пароля в течение короткого промежутка времени, а также наличие изменений в критических файлах системы.

Для разрабатываемой СОВ в качестве входных были выбраны следующие параметры:

1. ID протокола;
2. номер порта (хост);
3. номер порта (гость);
4. IP адрес (хост);
5. IP адрес (гость);
6. протокол ICMP;
7. нбъем передаваемых данных в байтах;
8. TCP-флаги.

В настоящее время в разрабатываемой системе используется алгоритм обучения, основанный на простой линейной регрессии. В качестве инструментария используется технология Google Drive. Процесс был разбит на два этапа:

1. обучение с использованием шаблонов Bro (создание паттернов нейронной сети);
2. обучение на основании паттернов, созданных на первом этапе.

Всего на первом этапе было использовано 260 вариантов входных данных с датчиков Bro. Этот этап, в силу ограничений Google Drive, проводился в две стадии. Результаты первого этапа приведены в таблице 2.1.

Таблица 2.1

Результаты первого этапа обучения сети

Стадии первого этапа	Количество входных данных с датчиков Bro	Количество созданных паттернов
1 стадия	143	26
2 стадия	117	25
Всего	260	51

После первого этапа обучения результат прогнозирования угроз был равен 82%.

Пример паттерна нейронной сети после второго этапа обучения приведен в таблице 2.2.

После второго этапа обучения результат достиг 96%.

Время обучения сети зависит от мощности вычислителя. В нашем случае на первом этапе время обучения составило 6 дней, а на втором этапе

всего 51 минуту. Обучение сети по одному шаблону составляет примерно 1 минуту, но нейронная сеть воспроизводит процесс обучения до 400 раз.

Таблица 2.2

Пример паттерна нейронной сети

№ п/п	Атрибут системы	Значение атрибута	Количество векторов атаки
1	Доступ злоумышленника	да/нет	0
2	Источник угрозы	Внешний/внутренний	1
3	Уязвимость	1…9	9
4	Тип атаки	1…5	5
5	Возможные атаки на ресурсы	1…39	39
6	Правильность ввода	да/нет	2
7	Зависимости, определенные скриптами	1…6	6
8	Вывод закодирован	да/нет	0
9	Аутентификация злоумышленником пройдена	да/нет	0
10	Доступ к URL адресу	да/нет	2
11	Безопасность HTTP	да/нет	3
12	Ошибки	да/нет	0

Результаты обнаружения атак нейронной сетью после двухэтапного обучения приведены в таблице 2.3.

Таким образом, первые полученные результаты обучения СОВ, основанной на нейронной сети, дают весьма обнадеживающие результаты.

Таблица 2.3

Обнаружение атак обучаемой нейронной сетью

№ п/п	Шаблоны атак	Вероятность обнаружения атаки нейронной сетью	Ожидаемый результат
1	DDos	1.0000	1

Окончание табл. 2.3

№ п/п	Шаблоны атак	Вероятность обнаружения атаки нейронной сетью	Ожидаемый результат
2	Bruteforce	1.0000	1
3	Атаки с доступом IP	1.0000	1
4	Трояны	0.9992	1
5	rootkit	0.9621	1
6	Spam	1.0000	1
7	Фишинг	1.0000	1
8	Вишинг	1.0000	1
9	Человек посередине	0.9899	1
10	Атаки на пароли по словарю	1.0000	1
11	Случайные скачивания	1.0000	1
12	Вредоносное ПО	0.772	1
13	Межсайтинговый скриптинг XSS	1.0000	1
14	Sql-инъекции	1.0000	1
15	Шифрование данных	0.891	1
16	Спуфинг	1.0000	1
17	Фрод	1.0000	1

Для улучшения производительности системы и качества обнаружения атак следует добавлять новые правила, создавать шаблоны и последовательно обучать нейронную сеть. Кроме того, предполагается варьировать набор входных данных, опробовать другие алгоритмы машинного обучения и дополнить систему инструментарием анализа рисков [4]. Авторы также считают необходимым привести состав и функции СОВ в соответствии с требованиями регуляторов.

Литература:

1. Bro Teaching and Training [Электронный ресурс]. URL: https://www.bro.org/ – загл. с экрана (дата обращения 11.11.2017).
2. Denning D.E. An intrusion detection model // IEEE Trans. on Software Engineering, 1987. – SE-13. – P. 222-232.
3. Lunt Teresa F. Detecting Intruders in Computer Systems // 1993 Conference on Auditing and Computer Technology, SRI International.
4. Васильева И.Н. Управление рисками информационной безопасности: учебное пособие. – СПб.: Изд-во СПбГЭУ, 2016. – 177 с.
5. Горбань А.Н., Дунин-Барковский В.Л., Кирдин А.Н. Нейроинформатика. – Новосибирск: Наука. Сиб.отд., 1998. – 280с.

6. Калан Р. Основные концепции нейронных сетей: нер.с англ. – М.: Издательский дом «Вильямс», 2003. – 288 с.
7. Нестерук Г.Ф., Куприянов М.С., Нестерук Л.Г. О реализации интеллектуальных систем в нечетком и нейросетевом базисах //Сб. докл. VI Междунар. конф. SCM'2003. – СПб.: СПГЭТУ, 2003. – т. 1. – С. 330-333.

2.3. Поиск средне- и высокоуровневых уязвимостей в машинном коде компьютерных систем

Буйневич М.В., Израилов К.Е.

Стремительное развитие компьютерных систем привело к их внедрению во все сферы деятельности человека. Однако к положительным сторонам такого процесса, основной из которых является автоматизация, добавились и отрицательные – это угрозы информационной безопасности, связанные с делегированием традиционно ручного (человеческого) функционала на исполнение автомату. Дело в том, что обработка данных в компьютерных системах осуществляется программным кодом, работа которого может отличаться от декларированной; используя это обстоятельство, злоумышленник способен нарушить конфиденциальность, целостность и доступность обрабатываемой информации. Такой код считается содержащим уязвимости и является небезопасным (здесь и далее под программным кодом будем подразумевать код программы в одной из форм ее жизненного цикла: архитектурной, алгоритмической, исходной, ассемблерной, машинной и др.). Рассмотрим современные возможности по поиску уязвимостей в машинном коде компьютерных систем.

Актуальность задачи поиска средне- и высокоуровневых уязвимостей в машинном коде и подходы к ее решению

Обратим внимание на тенденцию безопасности программного кода за последние 15 лет. Согласно открытым базам (таким, как NVD, CVE, OSVDB и др.) ежегодно наблюдается рост уязвимостей, обнаруживаемых в программном обеспечении. График такого роста для ведущих производителей телекоммуникационного оборудования, составляющего значительную часть устройств обработки информации и характеризующегося ощутимыми последствиями от нарушений информационной безопасности [2], представлен на рис. 2.6.

В случае использования оборудования, содержащего программное обеспечение, импортного производства, программный код имеет вид машинного (далее – МК), гарантированное обеспечение безопасности которого является крайне сложной задачей, как с технической, так и научной стороны. Во-первых, существующие средства нейтрализации уязвимостей

в МК не обладают требуемым набором возможностей. А, во-вторых, неразвитость теоретической базы этой сферы информационной безопасности не позволяет готовить достаточное количество экспертов необходимой квалификации.

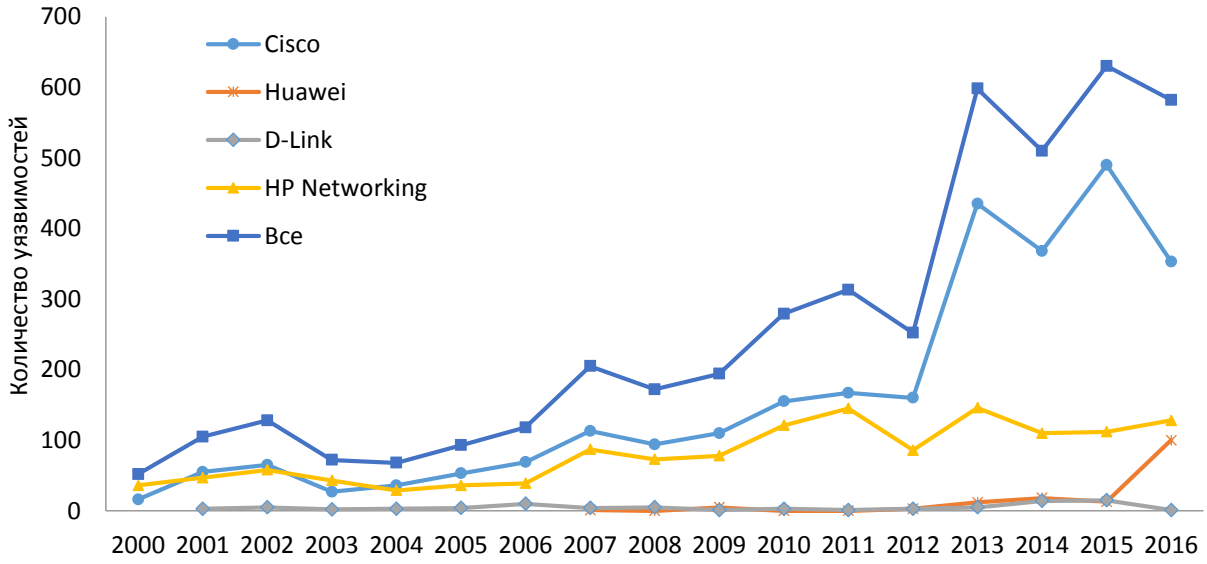


Рис. 2.6. Уязвимости в программном обеспечении телекоммуникационного оборудования ведущих производителей (согласно NVD) [18]

Состояние «импортозависимости» характерно для многих стран, в том числе и для России. Это подтверждается низкими показателями доли телекоммуникационного оборудования операторов связи и объема рынка программного обеспечения отечественного производства – около 10% и 20% соответственно [5]. Следовательно, использование заведомо небезопасного программного обеспечения импортного производства в большинстве случаев и в обозримом будущем будет неизбежным. Таким образом, существует проблема, заключающаяся в росте количества уязвимостей используемого программного обеспечения при ограниченности возможностей (количества и квалификации) экспертов информационной безопасности по их поиску.

Для разрешения проблемы безопасности программного кода можно выделить следующие шесть подходов ([8]), связанных с процессами создания МК любой компьютерной системы и обработки данных в ней.

1. Криптографическая защита информации. Несмотря на достаточную развитость криптографии и широкую применимость ее для обеспечения информационной безопасности (включая целый набор криптографических алгоритмов и протоколов), перспективным решением проблемы обоснованно можно считать только, так называемое, гомоморфное шифрование, позволяющее оперировать данными в зашифрованном виде.

2. Разработка безопасного исходного кода. Создание заведомо безопасного (то есть без уязвимостей) исходного кода на текущий момент считается лишь направлением теоретического исследования, которое, впрочем, может иметь практические результаты, например – понижение влияния человеческого фактора на появление уязвимостей, а также новые парадигмы программирования с концепцией, снижающей возможности такого появления.

3. Сборка исходного кода в безопасный. Многие средства сборки исходного кода в машинный имеют встроенные алгоритмы предупреждения разработчику о созданных им ошибках. Тем не менее, подход будет работать лишь в процессе разработки программного обеспечения, что не гарантирует безопасность от дальнейшей модификации МК.

4. Поиск уязвимостей в исходном коде. Непосредственное исследование исходного кода (в том числе и автоматическими средствами анализа) на предмет наличия в нем уязвимостей может стать разрешением проблемы. Основным препятствием этому подходу является тот факт, что подавляющее большинство производителей не поставляют открытый код своего программного обеспечения.

5. поиск уязвимостей в исполняемом коде. Исследование МК на предмет наличия уязвимостей является более универсальным и гипотетически более результативным, поскольку анализируется конечный вид программного обеспечения. Однако, как было отмечено выше, откровенно слабая научная база и на данный момент высокая трудоемкость подобного рода анализа не позволяют говорить о полноценном решении проблемы.

6. Создание безопасной среды выполнения программного кода. Идея создания операционной среды, препятствующей эксплуатации уязвимостей, уже сейчас частично реализована в виртуальных машинах (таких, как Java Virtual Machine и Common Language Runtime). Однако низкая производительность таких машин и противодействие лишь тривиальным уязвимостям не приводит к их повсеместному использованию.

Для сравнения возможностей подходов к обеспечению безопасности программного кода выделим следующие три типа уязвимости по их структурному уровню в коде:

- низкоуровневые (или вычислительные, далее – НУ), такие, как ошибки в операциях, структурах данных, доступе к ним и т. п. (например, выход за границы массива);
- среднеуровневые (или алгоритмические, далее – СУ), такие, как неверная реализация алгоритма подпрограммы, передачи входных и выходных параметров и т. п. (например, ошибка в условии вызова подпрограммы);

– высокоуровневые (или архитектурные, далее – ВУ), такие, как ошибки в архитектуре программной системы или концепции, нарушение общих принципов функционирования, безопасности и т.п. (например, использование слабых алгоритмов шифрования или возможность компрометации секретных ключей).

Если НУ, как правило, приводят к отдельным сбоям в работе компьютерной системы, а СУ нарушают логику алгоритмов, то ВУ ставят под угрозу безопасность всей системы в целом [3]. При этом наиболее простыми для обнаружения являются НУ, а наиболее сложными – ВУ.

Как можно видеть, применимость подходов к безопасности программного кода для указанных типов уязвимостей следующая. Первый оказывает противодействие всем типам уязвимостей, поскольку защищает исключительно обрабатываемую информацию, а не код. Второй в ближайшей теоретической перспективе сможет нейтрализовать лишь НУ и СУ, для ВУ же требуются специализированные научные исследования и еще более существенные результаты. Третий, являясь автоматическим средством обнаружения, применим только к формализуемым уязвимостям, а именно – НУ. Четвертый, подобный третьему, но расширенный привлечением ручного труда эксперта, может быть использован для поиска всех трех типов уязвимостей. Пятый позволяет обнаруживать все виды уязвимостей, хотя и имеет в разы более сложную трудоемкость, чем четвертый. Шестой позволит нейтрализовать лишь НУ, поскольку они имеют формализуемый эффект выполнения; остальные же типы уязвимостей без привлечения труда экспертов не обнаруживаемы.

В случае наличия МК, используемыми подходами могут быть первый, пятый и шестой. Применимость ко всем типам уязвимостей (и НУ, и СУ, и ВУ) выделяет только первый и пятый. Исходя из того, что первый эффективно решает лишь собственную область задач и не применим, как общее решения, наиболее перспективным можно считать пятый.

Поиск уязвимостей в исполняемом коде, который в большинстве случаев является машинным, теоретически позволяет найти все типы уязвимостей. Однако если и существуют средства автоматического поиска в нем НУ и некоторых СУ, то большинство СУ и все ВУ возможно обнаружить лишь ручным способом, поскольку само понятие уязвимости крайне субъективно и требует участия эксперта в каждом отдельном случае. Так как ручной поиск уязвимостей в МК крайне трудоемок, то одно из решений может лежать в области восстановления исходного представления машинного кода, подходящего для анализа экспертом. Рассмотрим возможности реализации данного подхода к безопасности программного кода путем решения следующей научной задачи – восстановление из МК представления, более близкого к архитектурному и алгоритмическому, и поэтому более подходящему эксперту информационной безопасности для ручного поиска СУ и ВУ.

Моделирование машинного кода

Наиболее близким способом преобразования МК в более высокоуровневую форму можно считать *декомпиляцию*, или восстановление исходного кода. Однако ее результаты нельзя считать удовлетворительными для эксперта, поскольку исходный код не отражает архитектуру МК, а алгоритмы в нем «смешаны» с вычислительными операциями. Для поиска же СУ и ВУ необходимо новое представление, подобное исходному, но отражающее именно алгоритмы и архитектуру МК, а также адаптированное для анализа экспертом с целью обнаружения уязвимостей.

Процесс получения такого представления имеет авторское название – *алгоритмизация* [11]. Возможность подобного преобразования обусловлена связью между низкоуровневой информацией в МК с высокоуровневыми элементами парадигм программирования (используемых при разработке исходного кода) посредством структурных метаданных.

Раскроем контекстный смысл введенного понятия. Под алгоритмизацией будем понимать восстановление организации и функционирования МК компьютерной системы в виде согласованной структуры (архитектуры) и алгоритмов программного кода – получение *алгоритмизированного представления*.

Для определения такой структуры необходимо выделение из МК данных специального типа (этимологически приводящих к появлению приставки мета-), при этом не хранящихся напрямую в нем. Такие данные, названные *структурными метаданными* (далее – СМД), состоят из следующей информации: модули и их взаимодействие, задающие архитектуру кода; взаимное использование подпрограмм и их данных, задающее строение модулей; алгоритмы и сигнатуры (имя, входные и выходные параметры), задающие функционал подпрограмм; управляющие структуры и логика их потока управления, задающие выполнение алгоритма.

Иерархия предлагаемых элементов СМД представлена в табличном виде (табл. 2.4).

Таблица 2.4

Иерархия элементов структурных метаданных

Родительский элемент	**Дочерние элементы**	**Связь дочерних элементов**
Архитектура	Модули	Взаимодействие модулей
Модуль	Подпрограммы, данные	Вызовы подпрограмм и использование ими разделяемых данных
Подпрограмма	Сигнатура (имя, входные/выходные параметры), алгоритм	Использование входных параметров алгоритмом для вычисления выходных
Алгоритм	Управляющие структуры	Логика потока управления

Рассмотрим типовые парадигмы программирования и СМД, которыми они оперируют.

Модульная парадигма программирования. Части МК, собранные из исходного кода отдельных файлов или их групп в директориях, в той или иной степени должны быть обособлены и в едином образе МК. Это является основным следствием назначения модулей – способа физического и логического деления всей совокупности программного кода. При этом, исходя из назначения каждого модуля, в качестве реализации обособленного функционала все внешнее взаимодействие с ним должно происходить по выделенному каналу вызовов – через его интерфейс. Таким образом, возможно выделение модулей в МК и их взаимосвязи, являющихся элементами СМД.

Процедурная парадигма программирования. Облик подпрограммы исходного кода должен оставаться также и в МК, поскольку подпрограмма относится к определяющему элементу сути (формально, топологии) алгоритма. С этой точки зрения в процессе получения МК из исходного суть алгоритмов остается неизменной, а меняется лишь их форма – с программно-языковой на бинарную, а также детали реализации – с понятных человеку на воспринимаемые выполняющим устройством. При этом, исходя из особенностей назначения подпрограммы, как многократно-используемого кода, она должна иметь в МК одну точку входа (с существующими переходами на нее извне) и, как правило, одну точку выхода в конце собственного кода. Подпрограммы могут быть вызваны из других таких же подпрограмм, анализ графа вызовов которых позволит восстановить логику их взаимодействия. Таким образом, возможно определение подпрограмм в МК и их взаимодействия, что является элементами СМД.

Структурная парадигма программирования. Все управляющие структуры должны найти свое отражение и в МК, поскольку они, как и подпрограммы, определяют суть или содержание алгоритма, которая остается неизменна при переходе от программно-языковой формы к бинарной. Так, в преобладающем количестве наборов инструкций процессоров существуют аналогичные им сущности: последовательно-выполняемые операции, оператор условного перехода и операторы цикла. Также, в МК оператор безусловного перехода GOTO имеет свое точное отражение, равно как и оператор вызова подпрограмм Таким образом, возможно определение управляющих структур в МК и построение по ним логики алгоритма, являющихся элементами СМД.

Императивная парадигма программирования. Никаких СМД напрямую в МК не создается. Тем не менее, отдельные вычисления могут быть использованы, как для уточнения метаданных структурной парадигмы – значений

условий для переходов, так и процедурной – значений возвращаемых значений и их связей с параметрами функций. Таким образом, возможно определение входных и выходных параметров подпрограмм – то есть частичной сигнатуры подпрограммы, являющейся элементом СМД.

Итак, высокоуровневые элементы каждой из парадигм программирования могут быть сопоставлены с введенными СМД.

Рассмотрим области выполнения и данных в ассемблерном коде, полученном из МК на предмет содержащихся в них СМД. В любом МК существует область, содержащая код выполнения – те инструкции процессора, которые непосредственно реализуют действия программного кода и используются конечным устройством для выполнения алгоритмов. В интересах выделения СМД в ассемблерном коде может быть найдена следующая информация:

- деление на подпрограммы;
- инструкции вычислений;
- инструкции безусловного перехода;
- инструкции условного перехода;
- инструкции цикла;
- инструкции вызова подпрограммы;
- код данных.

Поскольку любой нетривиальный МК в работе своих алгоритмов использует глобальные данные простых типов (константы, глобальные переменные) и составных типов (массивы и структуры; объединения являются вырожденным случаем), то часто для их хранения в коде отводится отдельная область – код данных. Анализ размещения глобальных переменных в коде данных и их использования в подпрограммах позволяет, во-первых, определить их семантический тип, а, во-вторых – связь между алгоритмами подпрограмм посредством «разделяемых» (shared) данных.

Низкоуровневые элементы МК могут быть сопоставлены с введенными СМД. Можно установить связь между элементами типовых парадигм программирования и МК через СМД (рис. 2.7). На всех структурных уровнях – вычисления, алгоритмов и архитектуры, по низкоуровневой информации из МК можно получить высокоуровневую информацию, связанную с исходным кодом. Отметим, что на тех же уровнях располагается каждая из введенных типов уязвимостей.

В интересах алгоритмизации полученную связь целесообразно отразить в соответствующей модели МК. Классическая линейная модель (L-модель) описывает МК в достаточно простом виде, не отражающем его особенности и их взаимосвязь с более высокоуровневыми представлениями; она также не предназначена для отражения каких-либо уязвимостей.

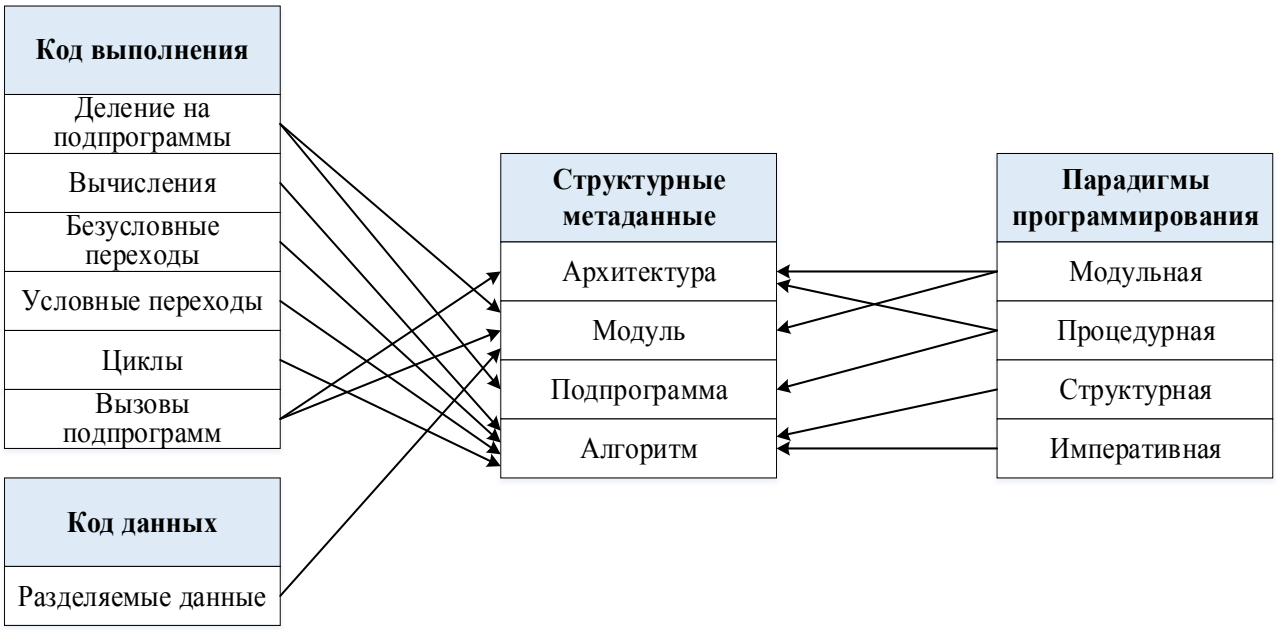


Рис. 2.7. Взаимосвязь между парадигмами программирования и машинным кодом через структурные метаданные

Взаимосвязь МК с парадигмами программирования, СМД и уровнями уязвимостей позволяет создать новую структурную модель – S-модель [9, 10], которая может быть использована для непосредственной алгоритмизации кода и последующего поиска уязвимостей (рис. 2.8).

S-модель позволяет формулировать научно-обоснованные требования к решению задач, связанных с восстановлением архитектуры и алгоритмов МК, одной из которых является создание метода и средства алгоритмизации.

Алгоритмизации машинного кода

Для проведения алгоритмизации МК с последующим поиском в нем уязвимостей (основной акцент среди которых делается на наиболее субъективные – СУ и ВУ) предлагается использовать авторский метод (далее – Метод) [6]. Суть Метода состоит в «S»-моделировании экземпляра МК и построении его алгоритмизированного «С»-подобного представления. Приведем описание Метода, включая средство его автоматизации и форматы основных используемых данных.

Метод состоит из 5 этапов (рис. 2.9).

На первом этапе производится дизассемблирование МК и получение его ассемблерного кода средствами продукта IDA Pro с применением скриптов. Данное программное средство является достаточно распространенным дизассемблером с богатыми графическими возможностями по анализу кода. Синтаксис ассемблерного кода расширен для получения возможности внесения будущих корректировок в процесс алгоритмизации.

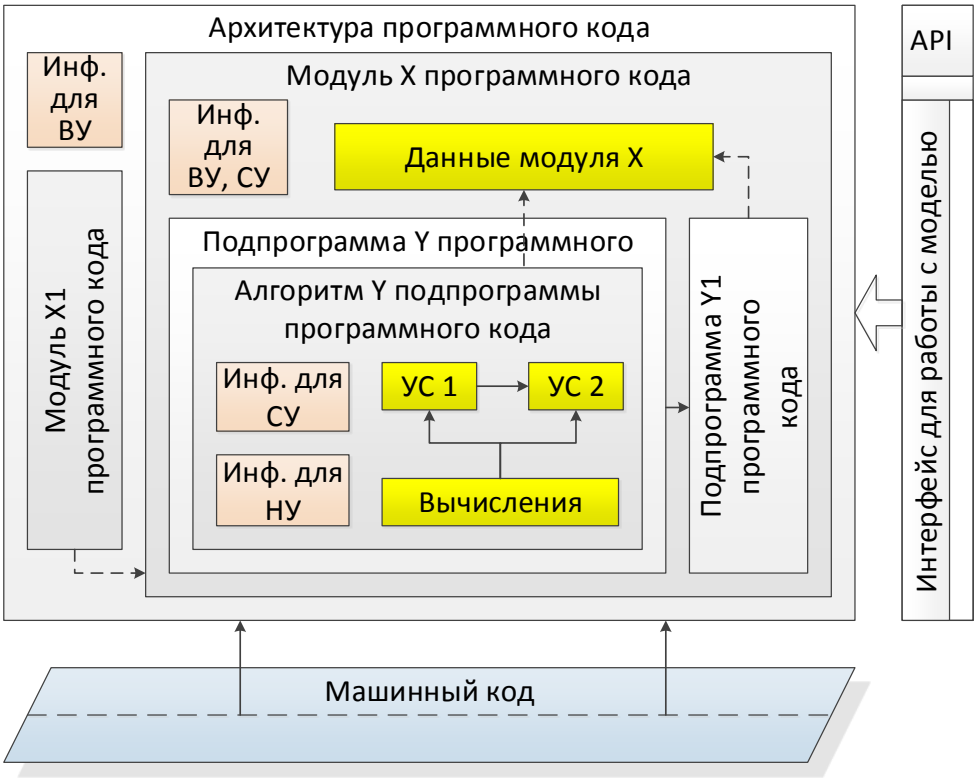


Рис. 2.8. Структурная S-модель машинного кода

На втором этапе полученный ассемблерный код алгоритмизируется специальным программным средством (далее – Утилита), последовательно выполняющим построение внутреннего представления ассемблерного кода, обработку представления со сбором и систематизацией информации об СМД и уязвимостях, а также создание внутреннего представления алгоритмов и архитектуры, генерируя его в текстовом виде.

На третьем этапе текстовое описание алгоритмов и архитектуры анализируется экспертом на предмет адекватности и удовлетворительности для последующего поиска уязвимостей.

На четвертом этапе (в случае недостаточно подходящего описания алгоритмов) эксперт производит корректировки процесса алгоритмизации, добавляя их в расширенный синтаксис ассемблерного кода, и повторяет запуск Утилиты.

В случае удовлетворительности описания алгоритмов производится их ручная гармонизация (пятый этап), т.е. доведение до вида, наиболее подходящего для будущего ручного поиска уязвимостей.

Для алгоритмизации МК разработано программное средство – Утилита [4], на которую было получено свидетельство о государственной регистрации программы для ЭВМ [15]. Поскольку очевидна близость цели и функционала Утилиты к подходам, используемым при компиляции и декомпиляции, то ее архитектуру целесообразно проектировать аналогичным

образом. Основополагающим элементом в архитектуре является S-модель, вокруг которой строится остальной функционал. Утилита имеет вид консольного приложения, а все взаимодействия с ней пользователя сведены до предоставления входных данных и анализа выходных.

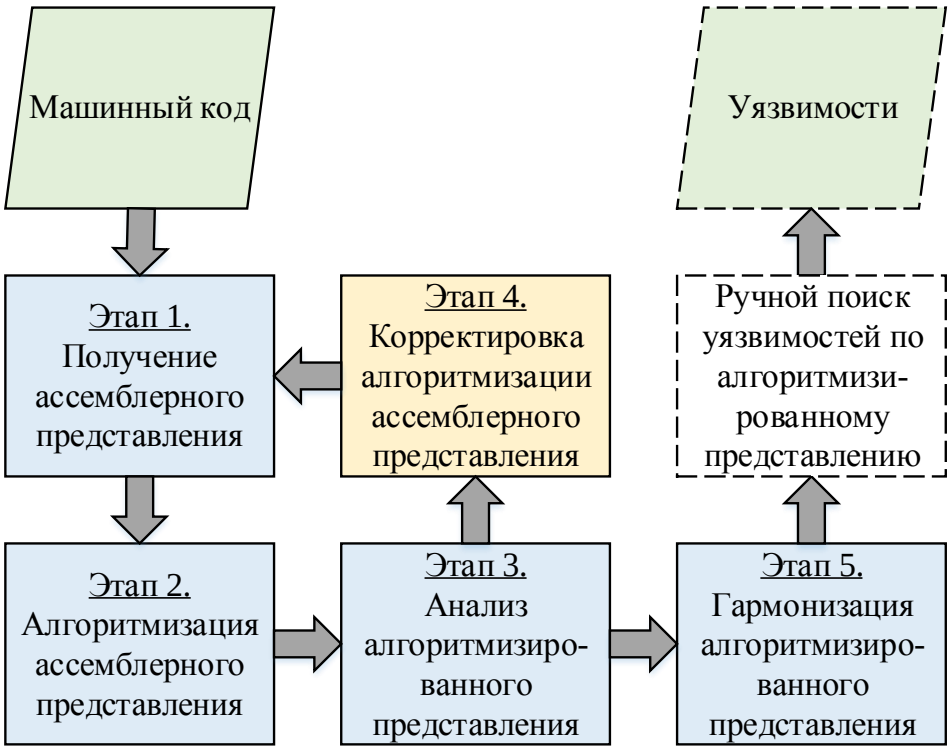


Рис. 2.9. Метод алгоритмизации машинного кода

Принцип работы Утилиты может сводиться к последовательному выполнению действий согласно этапам Метода. S-модель в такой архитектуре задается совокупностью различных внутренних структур – списков, хэшей и графов, на которых определены алгоритмы модулей Утилиты. Все модули представляют собой независимые единицы выполнения, определяющие ее функциональную архитектуру [7]. Обмен между модулями [17] на различных стадиях работы осуществляется посредством внутренних представлений Утилиты, определяющих ее информационную архитектуру [16]. Положительные результаты проведенного тестирования средства алгоритмизации [12] позволяют говорить об успешности выбранной архитектуры и ее программной реализации.

В Методе получение ассемблерного кода из МК производится с помощью широко распространенного дизассемблера из состава IDA Pro, поэтому входной синтаксис подобен используемому в продукте. А поскольку Утилита на вход принимает помимо ассемблерного кода АК еще и корректировки алгоритмизации, то синтаксис расширен поддержкой специальных конструкций:

a) для подпрограмм – «C»-подобный стиль декларации, имя и адрес точки входа, имена и регистры аргументов, регистры возвращаемых значений;

b) для глобальных переменных – имена и адреса глобальных переменных.

Утилита на выходе генерирует текстовое описание алгоритмов и архитектуры ассемблерного кода АК с использованием специального синтаксиса, подобного языку С, но имеющего ряд следующих отличий, повышающих компактность и воспринимаемость кода [13]:

a) для подпрограмм – «умный» формат имен переменных, регистры размещения переменных, операция битового доступа к данным, универсальные циклы, многоуровневый выход из циклов, возврат нескольких значений из подпрограммы;

b) для глобальных переменных – имена и адреса переменных;

c) для модулей – группирование подпрограмм, привязка разделяемых данных;

d) для корректировок алгоритмизации – расширяемый набор специальных настроек.

МК подпрограммы абсолютно не подходит для анализа экспертом. Однако после применения Утилиты полученное алгоритмизированное представление имеет синтаксис, близкий к языку программирования С, в котором отсутствуют несущественные для понимания алгоритма конструкции (в данном случае, типы значений). Такой код является более предпочтительным для эксперта с точки зрения поиска СУ и ВУ. Рассмотрим более подробно поиск данных типов уязвимостей с применением алгоритмизации.

Поиск средне- и высокоуровневых уязвимостей в машинном коде с применением алгоритмизации

Как подчеркивалось ранее, основным предназначением алгоритмизации является именно восстановление алгоритмов и архитектуры МК; но затем, полученное таким образом представление может быть использовано экспертом для ручного поиска уязвимостей. Для оценки эффективности поиск уязвимостей с применением алгоритмизации может быть применена соответствующая авторская методика [1]. Примеры применения Метода в интересах безопасности программного кода практически любой компьютерной системы представлены далее.

Пример поиска среднеуровневой уязвимости. Хотя основной сложностью поиска СУ и ВУ является задача их формализации, тем не менее, некоторые уязвимости могут быть обнаружены автоматически уже в процессе алгоритмизации.

Предположим изначально имеется функция аутентификации, сравнивающая введенный пароль с заданным (для простоты пароли заданы числовыми идентификаторами) – функция возвращает число 0x1 («True») или

0x0 («False») в зависимости от того, совпадают пароли или нет. Однако в результате злонамеренных действий в код (а именно, в начало тела функции) может быть встроена программная закладка, приводящая к тому, что будет возвращаться значение 0x1; таким образом, аутентификация всегда будет успешной. Данные действия злоумышленника приведут к угрозе нарушения конфиденциальности информации – в систему будет осуществлен доступ неавторизованного пользователя.

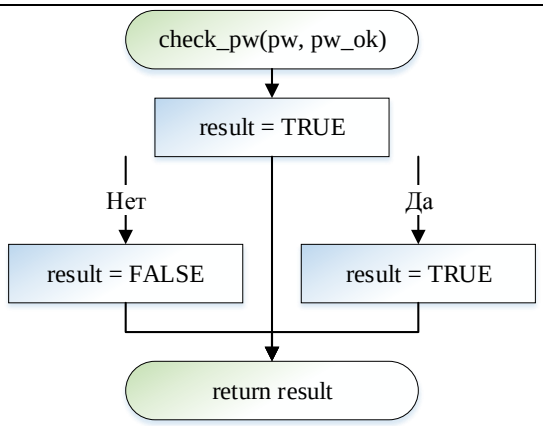
Результаты алгоритмизации МК без закладки и с закладкой приведены в табл. 2.5. По результатам применения Утилиты видно, что она не только корректно восстановила разрушенный алгоритм, но и проинформировала об этом эксперта: «ATTENTION!!! Possible, destruction of the structure» («ВНИМАНИЕ! Возможно, разрушение структуры»).

Пример поиска среднеуровневой уязвимости. Осуществим поиск СУ в коде подпрограммы, осуществляющей доступ к области памяти. Предположим, подпрограмма *hach_add_sec()* в качестве входного аргумента использует переменную (передаваемую через регистр *r3*) а затем увеличивает на ее значение глобальную переменную *sec* (расположенную по адресу 0x1000). Суть потенциальной СУ состоит в том, что глобальная переменная *sec* по логике разработчиков ПО не может быть модифицирована из кода. Тем не менее, подпрограмма *hack_add_sec()* делает обратное.

Таблица 2.5

Пример применения программного средства алгоритмизации к машинному коду функции проверки введенного пароля

Результат алгоритмизации МК без закладки	Алгоритм МК со встроенной закладкой	Результат алгоритмизации МК с закладкой

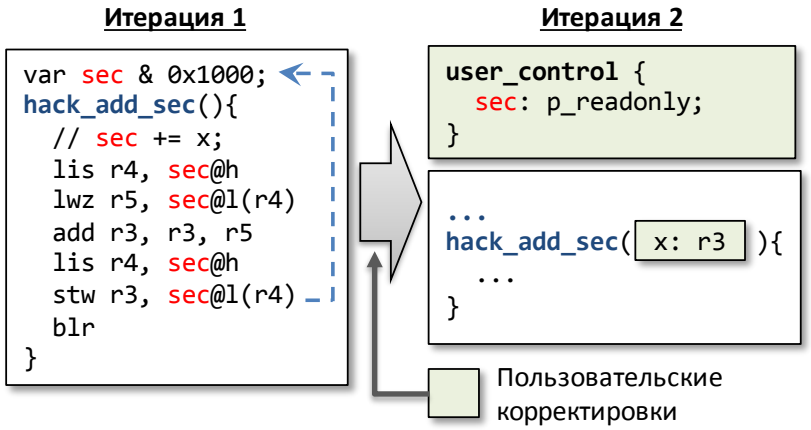
check_pw(pw, pw_ok)
{
if (pw == pw_ok)
{
result = 0x0;
} else {
result = 0x1;
}
return (result);
}
		check_pw(pw, pw_ok)
		{
		/* ATTENTION!!! Possible, destruction of the structure. */
		return (0x1);
		}

Применение Метода и Утилиты состоит из двух итераций. На первой эксперт по алгоритмизированному представлению принимает решение относительно способа доступа к переменной. Для этого он вносит

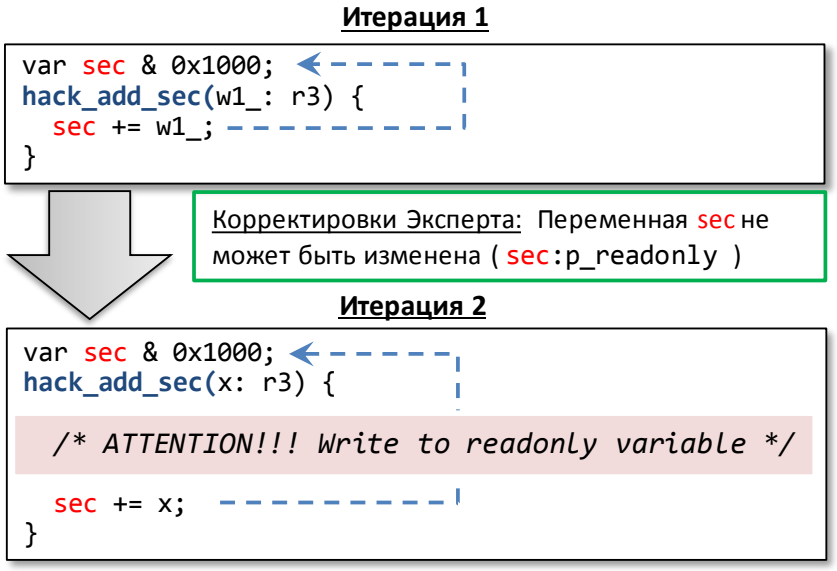
корректировки к алгоритмизации, указывающие, что переменная *sec* не может модифицироваться:

```
user_control {
   sec: p_readonly;
}
```

На второй итерации Утилита, учитывая корректировки эксперта, восстанавливает алгоритм. Итерационный процесс и результаты работы приведены на рис. 2.10. Как видно, Утилита сигнализирует о потенциальной уязвимости – нарушении доступа к переменной – с помощью комментария в алгоритмизированном представлении: «ATTENTION!!! Write to readonly variable» («ВНИМАНИЕ!!! Запись в переменную, доступную только для чтения»).



а)

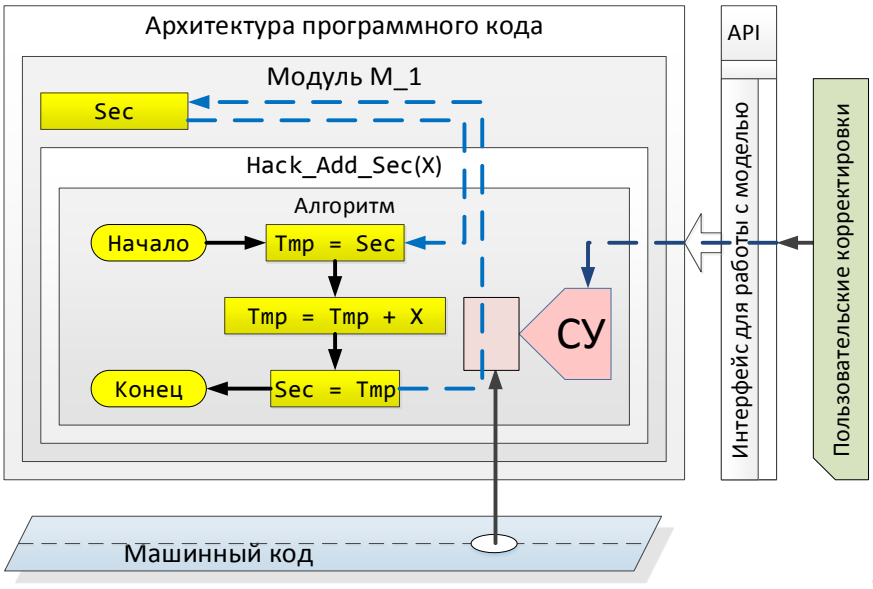


б)

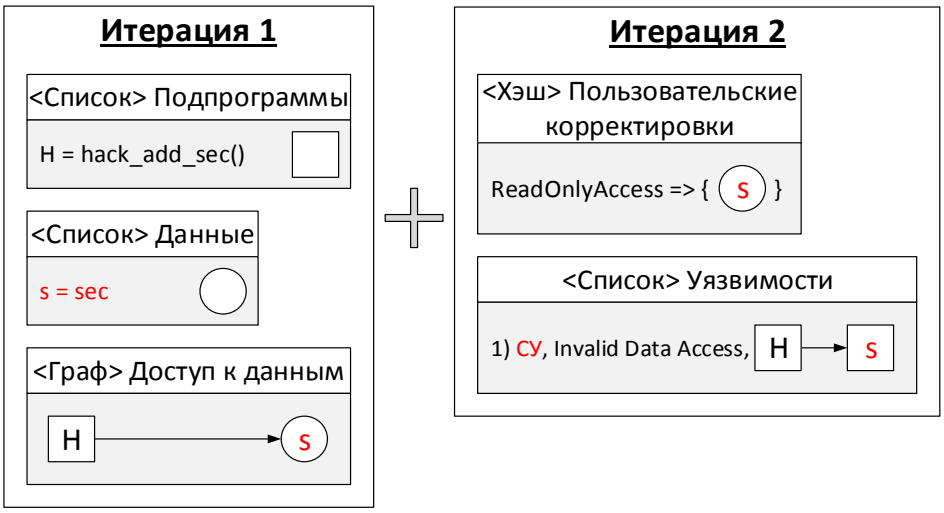
Рис. 2.10. Итеративный процесс и результаты алгоритмизации для машинного кода подпрограммы инкрементации глобальной переменной
а) ассемблерное представление; б) алгоритмизированное представление

В ходе работы Утилиты создается S-модель МК в виде совокупности внутренних данных (рис. 2.11).

На первой итерации внутреннее представление S-модели в Утилите задается с помощью списков подпрограмм МК (*H = hack_add_sec()*), глобальных переменных (*s = sec*) и их взаимосвязи через граф доступа (*H → S*). На второй итерации в представление модели добавляются пользовательские корректировки относительно допустимого способа доступа к переменной (*ReadOnlyAccess → S*), а также синтезируется информация об уязвимостях (*СУ, Invalid Data Access, H → S*). Таким образом, Метод и входящая в его состав Утилита при минимальном участии человека произвела моделирование МК и проинформировала о потенциальной СУ.



а)



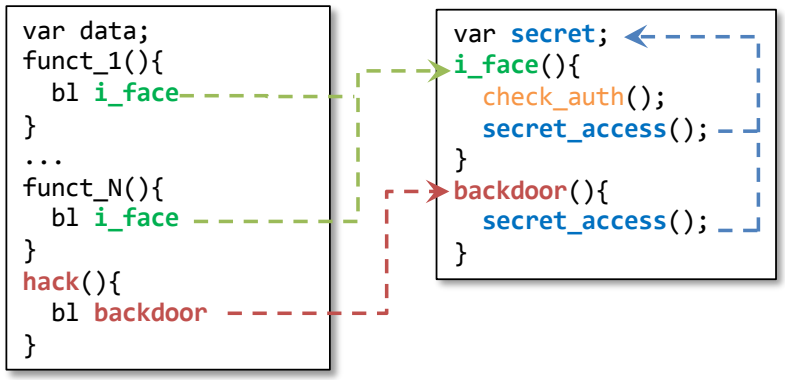
б)

Рис. 2.11. S-модель машинного кода подпрограммы инкрементации глобальной переменной (а) и ее внутреннее представление в Утилите (б)

Пример поиска высокоуровневой уязвимости. Осуществим поиск ВУ в коде взаимодействия двух модулей посредством программного интерфейса для доступа к конфиденциальным данным. Предположим первый модуль содержит собственную неиспользуемую переменную *data* и набор подпрограмм: *funct_1()* … *funct_N()* – вызывающих интерфейс *i_face()* второго модуля; *hack()* – вызывающую подпрограмму *backdoor()* второго модуля. Второй модуль содержит собственную переменную с конфиденциальными данными *secret* и набор подпрограмм: *i_face()* – предоставляющую стандартный интерфейс для внешних вызовов, а также осуществляющую проверку аутентификации и доступ к переменной *secret*; *backdoor()* – осуществляющую доступ к переменной *secret* без проверки аутентификации. Суть потенциальной ВУ состоит в том, что подпрограмма первого модуля *hack()* запрашивает конфиденциальные данные второго модуля из переменной *secret* с помощью подпрограммы *backdoor()*, не являющейся интерфейсной.

Результаты работы Утилиты приведены на рис. 2.12. Видно, что Утилита сообщила эксперту о количестве интерфейсных вызовов второго модуля (*i_face()* – N раз, *backdoor()* – 1 раз):

```
// i_face() – Interface with N calls
// backdoor() – Interface with 1 calls
```



а)

```
module M_1 {
  var data;
  funct_1(){ i_face(); }
  ...
  funct_N(){ i_face(); }
  hack(){ backdoor(); }
}

module M_2 {
  var secret;
  // i_face() - Interface with N calls
  // backdoor() - Interface with 1 calls
  interface(){ check_auth(); secret_access(); }
  backdoor(){ secret_access(); }
}
```

б)

Рис. 2.12. Результаты алгоритмизации взаимодействия двух модулей:
а) ассемблерное представление; б) алгоритмизированное представление

Таким образом, эксперт может обоснованно предположить наличие ВУ – «бэкдор» доступ первого модуля ко второму в обход интерфейса посредством вызова подпрограммы *backdoor()*.

В ходе работы Утилиты также создается S-модель МК в виде совокупности внутренних данных (рис. 2.13).

В процессе алгоритмизации внутреннее представление S-модели в Утилите задается с помощью списков подпрограмм МК (*1 = funct_1() … N = = funct_N(), H = hack(), I = i_face(), C = check_access(), S = secret_access(), B = = back_door()*), глобальных переменных (*d = data, s = secret*), их взаимосвязи через граф доступа (*[1, …, N, H] → d; [I, B, C, S] → s*), взаимных вызовов подпрограмм (*[1, …, N] → I → [C, S]; H → B → S*), а также хэша с группировкой логических элементов кода в модули (*M_1 → {data, funct_1(), … , funct_N(), hack()}, M_2 → {secret, i_face(), check_auth(), secret_access(), backdoor()}*).

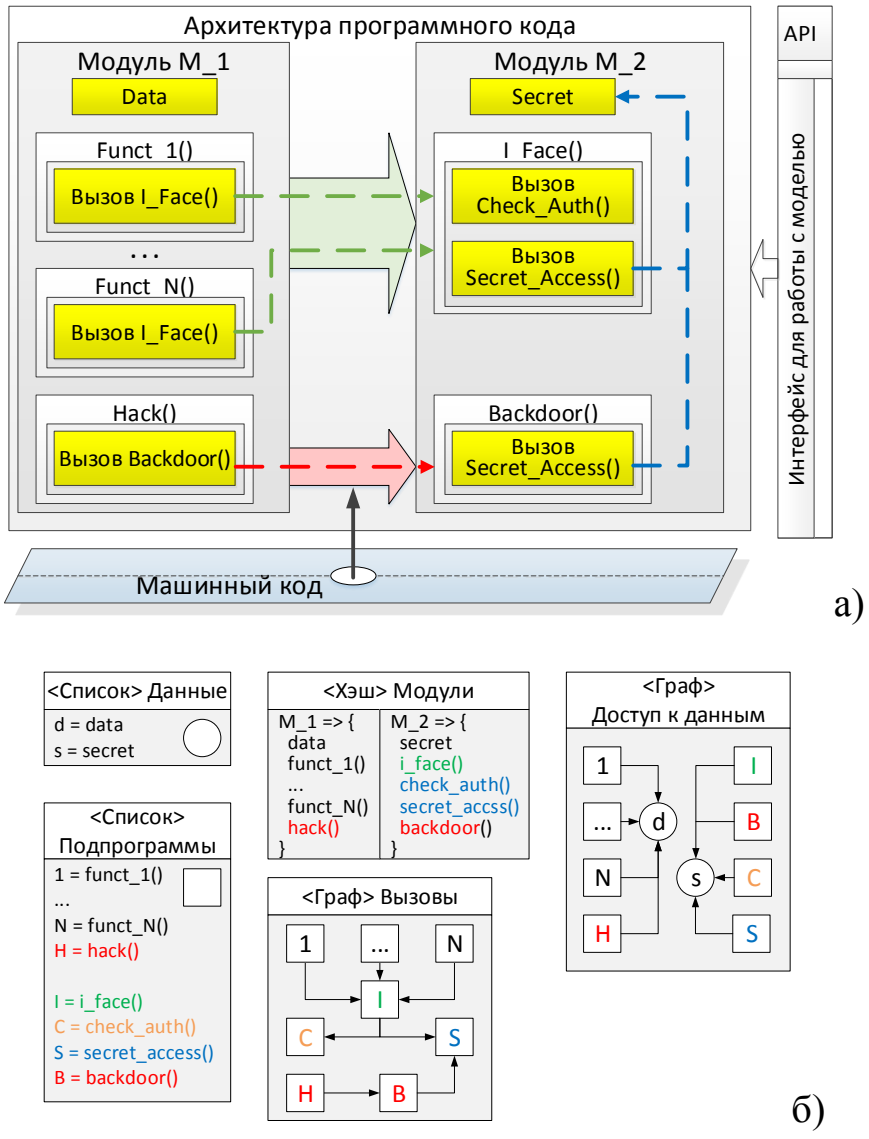


Рис. 2.13. S-модель машинного кода взаимодействия двух модулей (а) и ее внутреннее представление в Утилите (б)

Таким образом, Метод и входящая в его состав Утилита при помощи моделирования МК осуществила базовое восстановление его архитектуры и предоставила информацию, использованную экспертом для определения потенциальной ВУ.

Дальнейшие научные исследования и развитие предложенного метода поиска СУ и ВУ в МК компьютерных систем могут быть следующими:

1. реализация интерактивной среды визуализации ассемблерного кода, синхронизированной с его архитектурой и алгоритмами, что позволит проводить «глубокий» ручной поиск уязвимостей в МК;
2. создание механизма сравнения логики работы МК для обнаружения внесенных уязвимостей, что существенно повысит эффективность плановой проверки неизменности основного функционала программного кода;
3. разработка специализированных алгоритмов для обнаружения уязвимостей для автоматизации поиска СУ и ВУ;
4. формализация процесса и критериев осознания программного кода человеком и развитие метрики понятности кода [14], что приведет к созданию формата представления алгоритмов и архитектуры, наиболее близких и понятных эксперту безопасности.

Литература:

1. Васильева А.Ю., Израилов К.Е., Рамазанов А.И. Укрупненная методика оценки эффективности автоматизированных средств, восстанавливающих исходный код в целях поиска уязвимостей // Вестник ИНЖЭКОНа. Серия: Технические науки. – 2013. – № 8 (67). – С. 107-109.
2. Буйневич М.В., Владыко А.Г., Доценко С.М. и др. Организационно-техническое обеспечение устойчивости функционирования и безопасности сети связи общего пользования. – СПб.: СПбГУТ, 2013. – 144 с.
3. Буйневич М.В., Владыко А.Г., Израилов К.Е., Щербаков О.В. Архитектурные уязвимости моделей телекоммуникационных сетей // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». – 2015. – № 4. – С. 86-93.
4. Буйневич М.В., Израилов К.Е. Автоматизированное средство алгоритмизации машинного кода телекоммуникационных устройств // Телекоммуникации. – 2013. – № 6. – С. 2-9.
5. Буйневич М.В., Израилов К.Е. Категориальный синтез и технологический анализ вариантов безопасного импортозамещения программного обеспечения телекоммуникационных устройств // Информационные технологии и телекоммуникации. – 2016. – Т. 4. – № 3. – С. 95-106.

6. Буйневич М.В., Израилов К.Е. Метод алгоритмизации машинного кода телекоммуникационных устройств // Телекоммуникации. – 2012. – № 12. – С. 2-6.
7. Буйневич М.В., Израилов К.Е. Утилита для поиска уязвимостей в программном обеспечении телекоммуникационных устройств методом алгоритмизации машинного кода. Часть 1. Функциональная архитектура // Информационные технологии и телекоммуникации. – 2016. – Т. 4. – № 1. – С. 115-130.
8. Буйневич М.В., Израилов К.Е., Мостович Д.И., Ярошенко А.Ю. Проблемные вопросы нейтрализации уязвимостей программного кода телекоммуникационных устройств // Проблемы управления рисками в техносфере. – 2016. – № 3(39). – С. 81-89.
9. Буйневич М.В., Израилов К.Е., Щербаков О.В. Модель машинного кода, специализированная для поиска уязвимостей // Вестник Воронежского института ГПС МЧС России. – 2014. – № 2 (11). – С. 46-51.
10. Буйневич М.В., Израилов К.Е., Щербаков О.В. Структурная модель машинного кода, специализированная для поиска уязвимостей в программном обеспечении автоматизированных систем управления // Проблемы управления рисками в техносфере. – 2014. – № 3(31). – С. 68-74.
11. Израилов К.Е. Алгоритмизация машинного кода телекоммуникационных устройств как стратегическое средство обеспечения информационной безопасности // Национальная безопасность и стратегическое планирование. – 2013. – № 2. – С. 28-36.
12. Израилов К.Е. Методика оценки эффективности средств алгоритмизации, используемых для поиска уязвимостей // Информатизация и связь. – 2014. – № 3. – С. 39-42.
13. Израилов К.Е. Расширение языка «С» для описания алгоритмов кода телекоммуникационных устройств // Информационные технологии и телекоммуникации. – 2013. – № 2. – С. 21-31.
14. Израилов К.Е. Система критериев оценки способов поиска уязвимостей и метрика понятности представления программного кода // Информатизация и связь. – 2017. – № 3. – С. 111-118.
15. Израилов К.Е. Утилита восстановления алгоритмов работы машинного кода: свидетельство о государственной регистрации программы для ЭВМ. – рег. № 2013618433. – 09 .09.2013.
16. Израилов К.Е. Утилита для поиска уязвимостей в программном обеспечении телекоммуникационных устройств методом алгоритмизации машинного кода. Часть 2. Информационная архитектура // Информационные технологии и телекоммуникации. – 2016. – Т. 4. – № 2. – С. 86-104.

17. Израилов К.Е., Покусов В.В. Утилита для поиска уязвимостей в программном обеспечении телекоммуникационных устройств методом алгоритмизации машинного кода. Часть 3. Модульно-алгоритмическая архитектура // Информационные технологии и телекоммуникации. – 2016. – Т.4. – № 4. – С. 104-121.
18. Национальная база данных уязвимостей (National Vulnerabilities Database) [сайт]. URL: https://nvd.nist.gov (дата обращения 11.11.2017).

ГЛАВА 3. АНАЛИЗ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И ОЦЕНКА ЭФФЕКТИВНОСТИ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

3.1. Метод категорирования информационных активов по требованиям безопасности с помощью анализа иерархий и кластерного анализа

Куватов В.И., Примакин А.И.

Задача обеспечения информационной безопасности средств вычислительной техники (СВТ) и автоматизированных систем (АС) по мере внедрения компьютеров во все стороны жизни личности, общества и государства становится все более важной. Поэтому раздел теории информационной безопасности, направленный на разработку систем защиты информации в СВТ и АС, является одним из наиболее востребованных разделов.

Наибольшую сложность при разработке системы защиты информации в СВТ и АС представляет задача формирования требований к защите каждого вида используемой в системе конфиденциальной информации, ее носителей и процессов обработки. Эта сложность в первую очередь связана с отсутствием формальных моделей сравнительной оценки величины требований конфиденциальности, целостности и доступности информации хранящейся и обрабатываемой в СВТ, АС.

Классический подход к решению этой задачи основан на учете только одного свойства безопасности информации – ее конфиденциальности [1]. Требования к обеспечению целостности и доступности информации, как правило, фигурируют опосредовано, в общих требованиях к СВТ и АС. Считается, что раз доступ к информации имеет узкий круг доверенных лиц, вероятность ее искажения (нарушения целостности) или блокирования (нарушения доступности) незначительна. Такой подход правомерен, если важность конфиденциальности много выше важности целостности и доступности.

В общем случае это не всегда так. В некоторых случаях, особенно при работе с коммерческой информацией, более важными могут оказаться такие качества как целостность и доступность.

Современный подход [1] предполагает, наряду с требованиями по обеспечению конфиденциальности информации, формирование требований по обеспечению целостности и доступности информации (последнее – только для санкционированных пользователей). Так для платежных документов самым важным свойством является свойство целостности (достоверности), следующим по важности является свойство доступности. Требование конфиденциальности для платежных документов является не столь важным.

Для каждого типа требований разрабатывается несколько уровней. Число уровней и их смысл могут различаться, важно чтобы они указывались конкретно, исходя из величины возможного ущерба, наносимого субъектам информационных отношений. Порядок определения уровня принято описывать следующим образом:

1. Составляется перечень типов информационных активов. Для этого данные делятся по тематике, по функциональному назначению, по одинаковости информационных технологий и др. признакам.

2. Для каждого типа данных и каждого свойства безопасности информации определяются: перечень и важность субъектов информационных отношений (важность незначительная, малая, средняя, высокая, очень высокая), интересы которых затрагиваются при нарушении данного свойства безопасности информации и уровень наносимого субъекту ущерба (незначительный, малый, средний, большой, очень большой).

3. Для каждого типа информационных активов с учетом значимости субъекта и уровней наносимого им ущерба устанавливается степень необходимой защищенности по каждому из свойств безопасности информации. В качестве уровней защищенности могут выступать: отсутствие требований по данному свойству, низкий, средний, высокий, очень высокий уровень требований. При равенстве значений важности для разных субъектов, в итоге выбирается максимальное значение уровня (см. таблицу 3.1).

Таблица 3.1

Пример представления результатов оценки требований к защищенности некоторого типа информации

Информационный актив	Требования по уровням защищенности		
	конфиденциальность	целостность	доступность
№1	нет	средняя	низкая
№2	низкая	низкая	высокая
№3	средняя	высокая	средняя
Итоговый уровень	средняя	высокая	высокая

Наиболее важный недостаток такого подхода связан с использованием ранговой шкалы для оценки величины каждого свойства. Ранги разных свойств информации одного типа не связаны между собой. Несвязанными являются также ранги одного свойства информации разных типов. Все это не дает возможности рационально распределять ресурсы между обеспечением различных свойств одного и того же типа информации, между одинаковыми свойствами разных типов информации [2].

В данной работе предлагается:

− метод сравнительной оценки важности субъектов, важности показателей конфиденциальности, целостности и доступности информации для каждого из субъектов;

− метод разделения субъектов информационных отношений на классы с одинаковыми требованиями к конфиденциальности, целостности и доступности информации.

В идейном отношении метод базируется на схеме анализа иерархий [2] и кластерный анализ [3]. Методику применения метода анализа иерархий рассмотрим на примере задачи оценки важности субъектов.

Задача оценки важности субъектов. Пусть требования безопасности к АС определяются интересами *n* субъектов. Объективно существуют, но нам неизвестны веса, характеризующие важность каждого из них, w_i. Эксперт на основе опыта и интуиции задал величины α_{ij}, характеризующие его суждения о преимуществе *i*-го субъекта над *j*-м. В идеале, когда эксперт действует безошибочно, $\alpha_{ij} = w_i / w_j$. В реальности это равенство, как правило, не соблюдается, и чем больше по абсолютному значению разность $\alpha_{ij} - w_i / w_j$, тем больше ошибки эксперта.

Сформируем матрицу парных сравнений (суждений) эксперта $A = \|\alpha_{ij}\|$, размерностью $n \times n$. Очевидно, что эта матрица будет обратно симметричная, т.е. для любых *i* и *j* будет иметь место соотношение $\alpha_{ij} = 1/\alpha_{ji}$, причем для всех *i* $\alpha_{ii} = 1$. Назовем матрицу A', для каждого элемента которой выполняются соотношения $\alpha'_{ij} = w_i / w_j$, полностью согласованной.

Для полностью согласованной матрицы получим, $\alpha'_{ij} \cdot w_j = (w_i / w_j) \cdot w_j = w_i$, *i, j*=1,..,*n*. В реальности это соотношение, как правило, выполняться не будет. Однако если предположить, что ошибки эксперта являются несмещенными, то есть их математическое ожидание равно отношению w_i / w_j, то, при достаточно большом *n* для обратно симметрической матрицы А будет иметь место равенство, $w_i = \sum_{j=1}^{n} w_j \cdot \alpha_{ij} \Big/ \lambda_{\max}$, где $\lambda_{\max} \approx n$, *i*=1,..,*n*. Перепишем это выражение в виде: $w_i \cdot \lambda_{\max} = \sum_{j=1}^{n} w_j \cdot \alpha_{ij}$. Получим известную из линейной алгебры задачу отыскания собственных векторов и собственных значений матрицы *A*. Как известно, собственные значения матрицы *A* находятся из матричного уравнения $A - \lambda \cdot E = 0$, где E – единичная матрица, причем количество собственных значений совпадает с размерностью матрицы, т.е. равно *n*.

Каждому собственному значению матрицы *A* соответствует бесконечное множество собственных векторов. В случае если ранг матрицы *A* равен *n*, то все собственные вектора, относящиеся к одному собственному значению матрицы коллинеарны, т.е. для любых двух векторов $W=(w_1,..,w_n)$ и $V=(v_1,..,v_n)$, принадлежащих одному λ, существует скалярная положительная величина *c*, такая, что $W=c\cdot V$. Поэтому вектор *W* всегда можно выбрать так, чтобы $\sum_{i=1}^{n} w_i = 1$. Но тогда компоненты данного вектора и будут представлять собой нормализованные веса важности субъектов информационных отношений. С учетом того, что данный вектор характеризует важность субъектов информационных отношений, обозначим его *Wc*.

В линейной алгебре доказано, что если матрица *A* полностью согласована, то все ее собственные значения равны 0, за исключением одного, равного *n*. Доказано также, что если элементы α_{ij} положительной обратно симметричной матрицы незначительно изменить, то ее собственные значения также изменятся незначительно.

С учетом двух приведенных фактов, по результатам решения уравнения $A-\lambda\cdot E=0$ можно судить о степени согласованности матрицы парных сравнений. Чем больше $\lambda_{\max}$ отличается от *n*, а остальные собственные значения – от нуля, тем более рассогласованной является эта матрица.

Степень рассогласования матрицы парных сравнений оценивают с помощью так называемого индекса согласованности, $\delta_{ис}=(\lambda_{\max}-n)/(n-1)$. Принято считать, что если индекс согласованности не превышает по абсолютному значению 0.1, то матрица парных сравнений согласована достаточно хорошо. В противном случае матрица считается рассогласованной и ее необходимо согласовать.

Простейший вариант повышения согласованности матрицы парных сравнений заключается в следующем. Вычислим вспомогательную матрицу в соответствии с выражением, $b_{ij}=\alpha_{ij}-w_i/w_j$, где w_i, w_j компоненты собственного вектора матрицы *A*. Найдем строку, для которой $b=\max_{i,j} b_{ij}$ и попросим эксперта откорректировать ее. Процесс необходимо продолжать, пока абсолютная величина индекса согласованности не окажется в заданных пределах.

Методика оценки важности субъектов:

1. Составить матрицу парных сравнений $A=\|\alpha_{ij}\|$.

2. Найти максимальное собственное значение матрицы парных сравнений $\lambda_{\max}$.

3. Рассчитать индекс согласованности $\delta_{ис}$ и, в случае необходимости, откорректировать матрицу парных сравнений.

4. Рассчитать нормализованный собственный вектор *W*, соответствующий максимальному собственному значению. Этот вектор и будет представлять нормализованные веса важности субъектов информационных отношений.

Задача оценки важности показателей конфиденциальности, целостности и доступности информационных активов. Пусть имеется *m* типов информационных активов, требующих обеспечения конфиденциальности, целостности и доступности. Известно, что требования по этим свойствам для разных субъектов могут различаться. Составим матрицу парных сравнений конфиденциальности размера $n \times n$ для информации первого типа (см. табл. 3.2).

Таблица 3.2

Матрица парных сравнений оценки требований конфиденциальности субъектов информационных отношений для информации первого типа

Конфиденциальность	Номер субъекта информационных отношений		
Номер субъекта информационных отношений	Субъект 1	……..	Субъект *n*
Субъект 1	α_{11}	α_{12}	α_{13}
……	α_{21}	α_{22}	α_{23}
Субъект *n*	α_{31}	α_{32}	α_{33}

Действуя по вышеприведенной методике, получим нормализованный вектор $W_{К1}$, характеризующий требования к конфиденциальности для каждого из *n* субъектов.

Рассуждая аналогично, получим нормализованный вектор $W_{Ц1}$ характеризующий относительную важность целостности и нормализованный вектор $W_{Д1}$ характеризующий относительную важность доступности информации первого типа для каждого из *n* субъектов.

Переходя к информационным активам второго,…, *m*-го типа, получим в конечном итоге множество из n векторов размерностью 4*m* (матрицу размера $4m \times n$), содержащую информацию о важности каждого субъекта, о требованиях конфиденциальности, целостности и доступности информации каждого типа по отношению к каждому из n субъектов. Теперь, воспользовавшись кластерным анализом [4], поделим это множество на классы с равными уровнями требований по конфиденциальности, целостности и доступности информации.

Метод анализа иерархий позволяет в данном случае получать более объективные экспертные оценки, чем оценки, получаемые другими методами, а кластерный анализ – разделить множество субъектов на близкие по требованиям конфиденциальности, целостности и доступности информации.

Литература:
1. Борисов М.А., Заводцев И.В., Чижов И.В. Основы программно-аппаратной защиты информации. – М.: URSS, 2015 – 412 с.
2. Куватов В.И., Синещук Ю.И., Синещук М.Ю. Проблемы выбора рационального состава системы защиты информации // Труды Юбилейной XV Санкт-Петербургской международной конференции «Региональная информатика «РИ-2016». – СПб, 2016. – С. 219-220.
3. Куватов В.И., Величко Г.А. Исследование операций. – Петродворец: ВВМУРЭ, 2005. – 425 с.
4. Неслухов Д.С. Использование кластерного и регрессионного анализа в изучении экономической деятельности судостроительных и судоремонтных предприятий // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №4 (2016) [электронный ресурс]. URL: http://naukovedenie.ru/PDF/78EVN416.pdf (дата обращения 11.11.2017).

3.2. Модели управления информационными рисками в системах условного доступа

Соколов Р.В.

Система условного доступа (Conditional Access System) предприятия связи – программно-аппаратный комплекс, предназначенный для ограничения доступа к платным кодированным цифровым спутниковым, кабельным, эфирным теле-и-радио каналам.

Далее рассматриваются принципы построения моделей управления информационными рисками в системах условного доступа (СУД) к коммуникационным каналам. Назначение моделей состоит в экономическом обосновании выбора вариантов организации системы противодействия информационным атакам на СУД, представляющих собой набор мероприятий противодействия атакам.

Предлагаются три модели условной оптимизации и модель безусловной оптимизации системы защиты СУД от информационных атак, представляющие собой модели линейного программирования с двоичными переменными.

Принципиальная структурная схема [1] систем условного доступа (СУД) представлена на рис. 3.1. Информационный поток владельца контента, прежде чем достичь абонентов проходит через СУД, задача которой – обеспечить прохождение информации к абонентам только при соблюдении определенных договорами условий по функциональности в соответствии с оплатой.

На пути информационного потока встречается модуль скремблирования и шифрования ESS (Encryption and Scrambling System), который использует ключи скремблирования, генерируемые модулем KMS (Key Management

System). За шифрованный информационный поток поступает в модуль обслуживания абонентов SAS (Subscriber Authorization System), который обеспечивает аутентификацию абонентов и защиту декодеров и смарт-карт от несанкционированного доступа. Непосредственно у абонентов находится модуль безопасности аппаратно – программного обеспечения декодера SRS (Secured Receiver System), который обеспечивает декодирование информационного потока, поступающего в приемную аппаратуру абонента.

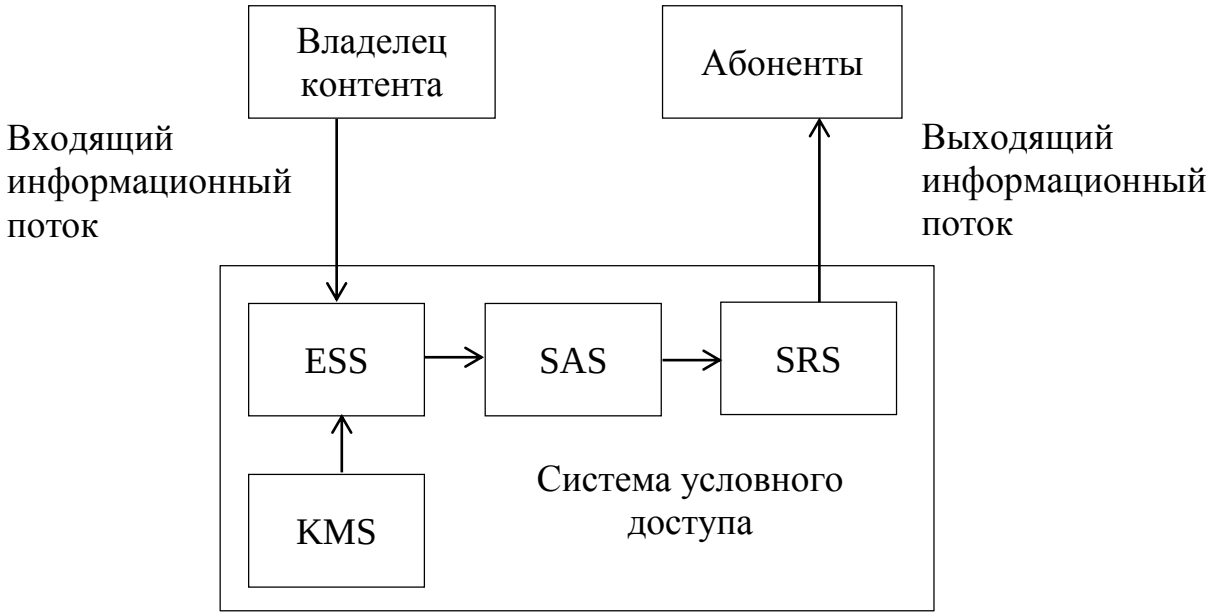


Рис. 3.1. Принципиальная структурная схема системы условного доступа

В настоящее время существуют десятки вариантов СУД, каждая из которых обслуживает десятки и сотни тысяч абонентов. СУД подразделяются на закрытые системы, использующие корпоративные стандарты шифрования и системы с единым механизмом скремблирования (Common Scrambling Algorithm), основанный на стандарте DVB (Digital Video Broadcasting). К числу расространненных СУД, используемых в России относятся следующие:

- DRECrypt (разработчик ООО «Цифра», Россия). Система условного доступа используется в России с 2004 г. Является одним из лидеров на рынке СУД, обслуживающем более 15 млн. абонентов. Внедрена у более чем 50 операторов платного телевидения РФ и стран СНГ. Поддерживает стандарт DVB.
- Viaccess (разработчик France Telecom, Франция). Ряд версий этой системы были взломаны и признаны неэффективными, например, Viaccess PC 2.3, Viaccess PC 2.5 и др. Версия ViaccessPC 5.0 не взломана и исполь-

зуется спутниковым оператором НТВ-Плюс в России и многими операторами в европейских странах. Относятся к числу наиболее распространенных систем условного доступа.

- Роскрипт (разработчик ФГУП НИИ Радио, Россия). Стандарт DVB обслуживает более 20 млн. абонентов.

СУД подвергаются взлому в целях несанкционированного доступа к информации, а также характеризуются возникновением инцидентов в обслуживании абонентов под влиянием информационных рисков.

Информационные атаки на СУД могут приводить к двум видам финансовых потерь:

- в случае взлома СУД возникает возможность пиратского использования контента без оплаты за него;
- в случае прерывания обслуживания абонентов возникают потери в абонентской плате.

В обоих случаях требуются затраты на восстановление нормальной работы СУД информационной системы.

Мероприятия по противодействию возможным информационным атакам на СУД включает в себя как специфические мероприятия по противодействию взлому, а также общие для всех информационным систем мероприятия.

К числу специфических мероприятий по противодействию взлому относятся: усложнение алгоритмов скремблирования, увеличение разрядности кодов, перепрограммирование смарт-карт, резервирование технического комплекса и др.

Методам защиты информации, которые могут быть применены в СУД, посвящен ряд работ [2-5, 9]. Однако экономико-математическая поддержка выбора варианта защиты информации раскрыта недостаточно. Поэтому управление информационными рисками в СУД с применением экономико-математических моделей и методов является актуальной темой исследования.

Принципы построения моделей управления информационными рисками

Рассмотрим принципы построения моделей управления информационными рисками в СУД [7].

Экономико-организационное назначение моделей. Назначение моделей состоит в экономическом обосновании выбора и организации варианта защиты СУД от информационных атак, представляющего собой набор мероприятий противодействия этим атакам.

1. Адекватность параметров и переменных моделей особенностям управления информационными рисками в СУД.

В состав параметров моделей должны входить параметры, характеризующие потоки информационных атак, мероприятий противодействия им, а также критерии и ограничения моделей. Переменные моделей должны характеризовать выбранный вариант системы информационной защищенности СУД.

2. Трехаспектный подход к оценке защищенности.

Наличие к качестве критериев и ограничений модели значений вероятностей взлома СУД, нарушения обслуживания абонентов за определенный интервал времени и стоимостных затрат на обеспечение информационной безопасности, то есть трехаспектный подход к оценке системы информационной защищенности.

3. Формирование и ведение базы данных моделей.

В качестве исходных данных целесообразно использовать базу данных видов угроз [10], их вероятностей и ущерба, полученных на основании статистической и прогнозной информации и возможных мероприятий по их противодействию.

4. Возможность учета лингвистической неопределенности в исходных данных.

Учет лингвистической неопределенности в части возникновения информационных атак, возможностей противодействия им и величин ущерба в случае наступления рисковых событий.

5. Одноразовость затрат на защитные мероприятия при его многоразовом использовании для противодействия разным видам атак. Отражение этого принципа в моделях обеспечивает снижение затрат на обеспечение информационной безопасности.

6. Выбор временного интервала в управлении информационными рисками.

Предполагается, что выбор достаточно малого временного интервала позволяет считать вероятность появления однократной атаки существенно выше вероятности появления атаки большей кратности, которыми можно пренебречь. При этом математическое ожидание суммы ущерба за определенный период времени представляется как сумма математических ожиданий ущерба за каждый интервал (подпериод) данного периода времени.

7. Классификация атак по принципу однородности.

Разделение атак (угроз) по принципу их однородности осуществляется с тем, чтобы мероприятия по противодействию рисковым событиям определенного класса можно было бы считать не дополняющими друг друга. В этом случае результат выбора наиболее эффективного мероприятия не будет улучшаться при наличии другого мероприятия, противодействующего атакам данного типа.

Рассмотрим возможные варианты моделей управления информационными рисками в СУД.

Модель условной оптимизации системы управления информационными рисками СУД по критерию минимума совокупной стоимости владения

Ниже представлена модель условной оптимизации системы управления информационными рисками в СУД по критерию минимума совокупной стоимости владения при ограничениях по вероятности взлома и нарушения обслуживания абонентов СУД.

$$\Pi = \sum_{m \in M} \Pi_m X_m \to \min$$

$$P_{\text{ВЗ}} = q_{\text{ВЗ}} \left(1 - \sum_{i \in I_{\text{ВЗ}}} K_i \sum_{m \in M} r_{im} Y_{im}\right) \le P_{\text{ВЗ.ДОП}}$$

$$P_{\text{об}} = P_{\text{ВЗ}} \left(1 - \sum_{i \in I_{\text{об}}} K_i \sum_{m \in M} r_{im} Y_{im}\right) \le P_{\text{ОБ.ДОП}}$$

$$\sum_{m \in M} Y_{im} = 1, \forall i \in I_{\text{ВЗ}}, \forall i \in I_{\text{об}} \quad (1)$$

$$X_m - Y_{im} \ge 0, \forall m \in M, \forall i \in I \quad (2)$$

$$X_m = \{0, 1\}, \forall m \in M$$

$$Y_{im} = \{0, 1\}, \forall m \in M, \forall i \in I$$

Здесь приняты следующие обозначения.

П – совокупная среднегодовая стоимость владения системой управления информационными рисками СУД;

$P_{\text{ВЗ}}, P_{\text{ВЗ.ДОП}}$ – соответственно вероятность и допустимая вероятность возникновения рискового события, связанного со взломом СУД, на протяжении интервала времени (подпериода);

$P_{\text{об}}, P_{\text{ОБ.ДОП}}$ – соответственно вероятность и допустимая вероятность нарушения обслуживания абонентов СУД на протяжении заданного интервала времени;

X_m – двоичная переменная, принимающая значение 1, если в СУД однократно или большее число раз используется m-й механизм для предотвращения угроз и принимает значение 0 в противном случае;

Y_{im} – двоичная переменная, принимающая значение 1, если для противодействия i-ой угрозе используется m-й механизм и принимающая значение 0 – в противном случае;

Π_m – совокупная среднегодовая стоимость владения m-ым механизмом противодействия угрозам;

$q_{\text{ВЗ}}$ – вероятность возникновения однократной атаки, угрожающей взлому СУД на протяжении заданного интервала времени (подпериода), при котором многократными атаками можно пренебречь;

$q_{об}$ – вероятность возникновения однократной атаки, угрожающей бесперебойному обслуживанию абонентов СУД на протяжении заданного интервала времени, при котором многократными атаками можно пренебречь;

r_{im} – вероятность отражения атаки i-го вида с помощью m-го механизма;

K_i – удельный вес атак i-го вида,

$$\sum_{i \in I_{ВЗ}} K_i = 1 \, , \sum_{i \in I_{об}} K_i = 1;$$

$I_{ВЗ}$ – множество видов атак, угрожающих взлому СУД;

$I_{об}$ – множество видов атак, угрожающих бесперебойному обслуживанию абонентов СУД;

I – множество видов угроз (атак) на СУД;

M – множество механизмов противодействия атакам.

Условие (1) означает, что для каждой угрозы должен быть выбран механизм противодействия, причем только один, так как результат выбора наиболее эффективного мероприятия не будет улучшаться другими мероприятиями для угроз данного класса при детальной классификации их по принципу однородности.

Неравенство (2) обеспечивает однократность учета затрат на многократное использование m-го механизма противодействия угрозам [6]. Это неравенство соответствует следующему выражению:

$$X_m = \begin{cases} 1 \, , \text{если} \sum_{i \in I} Y_{im} = 1, 2, 3 \ldots \\ 0 \, , \text{если} \sum_{i \in I} Y_{im} = 0. \end{cases}$$

На основе рассмотренной модели можно предложить модель взаимно – обратной задачи оптимизации, в которой критерии и ограничения меняются местами [8].

Модель условной оптимизации системы управления информационными рисками СУД по критерию минимума вероятности взлома СУД

Задача условной оптимизации, в которой в качестве критерия выступает минимизация вероятности взлома СУД на протяжении заданного интервала времени, имеет следующий вид.

$$P_{ВЗ} = q_{ВЗ} \left(1 - \sum_{i \in I_{ВЗ}} K_i \sum_{m \in M} r_{im} \, Y_{im} \right) \rightarrow \min$$

$$П = \sum_{m \in M} П_m \, X_m \leq П_{доп}$$

$$\mathrm{P}_{\text{об}} = q_{\text{об}} \left(1 - \sum_{i \in I_{\text{об}}} K_i \sum_{m \in M} r_{im} Y_{im} \right) \leq P_{\text{ОБ.ДОП}}$$

$$\sum_{m \in M} Y_{im} = 1 , \forall\, m \in M , \forall\, i \in I$$

$$X_m - Y_{im} \geq 0, \forall\, m \in M , \forall\, i \in I$$

$$X_m = \{ 0, 1 \}, \forall\, m \in M$$

$$Y_{im} = \{ 0, 1 \}, \forall\, m \in M , \forall\, i \in I,$$

где $\Pi_{\text{доп}}$ – допустимая совокупная среднегодовая стоимость владения системой управления информационными рисками СУД.

Модель условной оптимизации системы управления информационными рисками СУД по критерию минимума вероятности нарушения обслуживания абонентов СУД

Модель условной оптимизации, в которой в качестве критерия выбран минимум вероятности нарушения обслуживания абонентов СУД на протяжении выбранного интервала времени, имеет вид, аналогичный предыдущей модели.

$$\mathrm{P}_{\text{об}} = q_{\text{об}} \left(1 - \sum_{i \in I_{\text{об}}} K_i \sum_{m \in M} r_{im} Y_{im} \right) \rightarrow min$$

$$\Pi = \sum_{m \in M} \Pi_m X_m \leq \Pi_{\text{доп}}$$

$$P_{\text{ВЗ}} = q_{\text{ВЗ}} \left(1 - \sum_{i \in I_{\text{ВЗ}}} K_i \sum_{m \in M} r_{im} Y_{im} \right) \leq P_{\text{ВЗ.ДОП}}$$

$$\sum_{m \in M} Y_{im} = 1 , \forall\, m \in M , \forall\, i \in I$$

$$X_m - Y_{im} \geq 0, \forall\, m \in M , \forall\, i \in I$$

$$X_m = \{ 0, 1 \}, \forall\, m \in M$$

$$Y_{im} = \{ 0, 1 \}, \forall\, m \in M , \forall\, i \in I.$$

Модель безусловной оптимизации системы управления информационными рисками в СУД

Перейдем к построению модели безусловной оптимизации системы управления информационными рисками в СУД. В этой модели отсутствуют условия соблюдения допустимых вероятностей возникновения рисковых событий, связанных со взломом и нарушением обслуживания абонентов СУД. Вместо этих условий в целевую функцию минимизации совокупной среднегодовой стоимости владения системой управления информационными рисками в СУД добавляются значения оценок математических ограничений

среднегодовых потерь при возникновении рисковых событий. В качестве ограничений остаются лишь ограничения на значения независимых переменных задачи. Таким образом, модель безусловной оптимизации имеет следующий вид.

$$\Pi = \sum_{m \in M} \Pi_m X_m + q_{\text{ВЗ}} \left(1 - \sum_{i \in I_{\text{ВЗ}}} K_i \sum_{m \in M} r_{im} Y_{im}\right) L_{\text{ВЗ}} N_{\text{ВЗ}}$$

$$+ q_{\text{об}} \left(1 - \sum_{i \in I_{\text{об}}} K_i \sum_{m \in M} r_{im} Y_{im}\right) L_{\text{об}} N_{\text{об}} \rightarrow min$$

$$\sum_{m \in M} Y_{im} = 1, \forall\, i \in I_{\text{ВЗ}}, \forall\, i \in I_{\text{об}},$$

$$X_m - Y_{im} \geq 0, \forall\, m \in M, \forall\, i \in I,$$

$$X_m = \{0, 1\}, \forall\, m \in M,$$

$$Y_{im} = \{0, 1\}, \forall\, m \in M, \forall\, i \in I,$$

где $L_{\text{ВЗ}}$ – усредненные финансовые потери, связанные со взломом СУД;

$L_{\text{об}}$ – усредненные финансовые потери, связанные с нарушением договорных обязательств по обслуживанию абонентов СУД, в случае возникновения инцидента в работе СУД;

$N_{\text{ВЗ}}$ – количество выбранных интервалов (подпериодов) в течение года для оценки вероятности появления рисковых событий, связанных со взломом СУД;

$N_{\text{об}}$ – количество выбранных интервалов (подпериодов) в течение года для оценки вероятности появления рисковых событий, связанных с нарушением обслуживания абонентов СУД.

Длина интервалов времени выбирается достаточно малой с тем, чтобы вероятность однократного появления рискового события в течение выбранного интервала значительно превосходила вероятность многократного появления рисковых событий, и ею можно было пренебречь.

Оценка математического ожидания суммарных потерь при появлении рисковых событий в течение года учитывается как сумма математических ожиданий потерь при появлении отдельных рисковых событий.

Предложенные выше экономико-математические модели представляют собой модели линейного программирования с двоичными переменными. Для нахождения решения по этим моделям можно воспользоваться известными математическими пакетами, например пакетом MATLAB, Lindo, WinQSB и др. Нахождение решения следует искать в человеко-машинном диалоге. Целесообразно также использовать сценарный подход на основе оптимистического, пессимистического и наиболее вероятного сочетания исходных данных.

Предложенные модели можно использовать в качестве основы для нечетко – логического моделирования управления информационными рисками, учитывая лингвистическую неопределенность ряда исходных данных.

Результаты расчета по моделям могут быть использованы для страхования информационных рисков.

Литература:

1. Костин М. Системы условного доступа, 2004 [Электронный ресурс]. URL: http://www.telesputnik.ru/archive/109/article/62.html (дата обращения 10.10.2016).
2. Курило А.П. Обеспечение информационной безопасности бизнеса. – М.: БДЦ-пресс, 2005. – 512 с.
3. Михальчук С.А., Стельмашонок Е.В. Управление параметрами системы защиты информации в условиях их неопределенности на основе вероятностно-статистической модели // Комплексная безопасность бизнеса в условиях экономической нестабильности. Материалы научно-практической конференции. Министерство образования и науки РФ, СПбГЭУ, Кафедра вычислительных систем и программирования / ответственные редакторы: Е.В. Стельмашонок, С.Н. Максимов. – СПб: СПбГЭУ, 2014. – С. 115-118.
4. Симонов С.В. Современные концепции управления информационными рисками, 2003 [Электронный ресурс]. URL: http://www.pmprofy.ru/content/rus/85/850-article.asp (дата обращения 10.10.2016).
5. Стельмашонок Е.В., Тарзанов В.В., Соколовская С.А. Управление информационной безопасностью предприятия на основе сбалансированной системы показателей // Инновационные преобразования в производственной сфере. Сборник научных трудов международной научной конференции. Казань, 2012. – Киров, 2012. – с. 289-294.
6. Соколов Р.В. Проектирование информационных систем. – СПб.: СПБГИЭУ, 2012. – 334 с.
7. Соколов Р.В. Модели управления информационными рисками в системах условного доступа // Региональная информатика (РИ-2016). Юбилейная XV Санкт-Петербургская международная конференция «Региональная информатика (РИ-2016)». Санкт-Петербург, 26-28 октября 2016 г.: Материалы конференции. \СПОИСУ. – СПб., 2016. – С. 263-264.
8. Соколов Р.В., Николаев М.О. Экономико-математические модели управления информационными рисками на предприятиях связи // Известия Санкт-Петербургского государственного экономического университета – СПб., 2016. – № 6. – С. 88-93.

9. Черешин Д.С., Кононов А.А., Новицкий Е.Г., Цыгичко В.Н. Методика оценки рисков нарушения информационной безопасности в автоматизированных информационных системах. Препринт. – М.: Институт системного анализа РАН, 1999.
10. IT-Grundschutz Manual, BSI, «Руководство по защите ИТ для базового уровня защищенности» [Электронный ресурс]. URL: https://www.bsi.bund.de/EN/TheBSI/thebsi_node.html (дата обращения 10.10.2016).

3.3. Оценка рисков безопасности локальной сети с применением технологий нечеткого моделирования

Семенова С.О.

Одним из важнейших критериев, влияющих на успешное функционирование организации, является создание системы управления информационной безопасностью (СУИБ). Такая система позволит определить необходимую степень защиты информационных активов, а также своевременно предотвратить возможные инциденты безопасности. Построение СУИБ является трудоемким и многозадачным процессом, охватывающим не только анализ технических характеристик информационной системы, но и организационной структуры, должностных инструкций и политики безопасности, а также методов управления в организации. Согласно международному стандарту ISO/IEC 27001 (ГОСТ Р ИСО/МЭК 27001–2006) процессная модель СУИБ включает в себя несколько этапов: планирование, реализация, проверка, действие (ПРПД) [3].

На этапе планирования происходит выявление возможных рисков информационной безопасности. Для того чтобы оценка рисков всей информационной системы организации была наиболее полной, необходимо отдельно рассмотреть риски, связанные с безопасностью локальной вычислительной сети (ЛВС), так как она является наиболее распространенным средством коммуникаций.

Степень защищенности сети зависит от множества критериев, в том числе от масштабов организации и особенностей ее деятельности. Поэтому анализ рисков ЛВС для организаций носит индивидуальный характер. Тем не менее, можно выделить наиболее распространенные критерии, которые присущи всем ЛВС. Таким образом, вероятные риски информационной безопасности можно разделить на три группы:

- риски, связанные с аппаратными средствами;
- риски, связанные с программным обеспечением;
- риски, связанные с человеческим фактором.

Первая группа рисков включает в себя события, связанные с надежностью и отказоустойчивостью сетевого оборудования. Вторая – возможные риски в результате использования уязвимостей программного обеспечения. Несмотря на то, что процесс передачи информации по сети практически полностью автоматизирован, вероятность возникновения риска, связанного с человеческой ошибкой также необходимо учитывать, такие риски входят в третью группу.

Рассматривая приведенные выше группы рисков можно отметить, что невозможно дать точную оценку их влияния на состояние защищенности сети из-за трудоемкости составления всех возможных событий, приводящих к возникновению данных рисков. Поэтому для моделирования риска информационной безопасности ЛВС целесообразно использовать методы нечеткой логики [2]. Для построения модели, характеризующей степень влияния выбранных групп рисков на общее состояние защищенности сети, был выбран программный продукт «MATLAB» с дополнительным пакетом «Fuzzy Logic Toolbox» [1]. Интерфейс программы представлен на рис. 3.2.

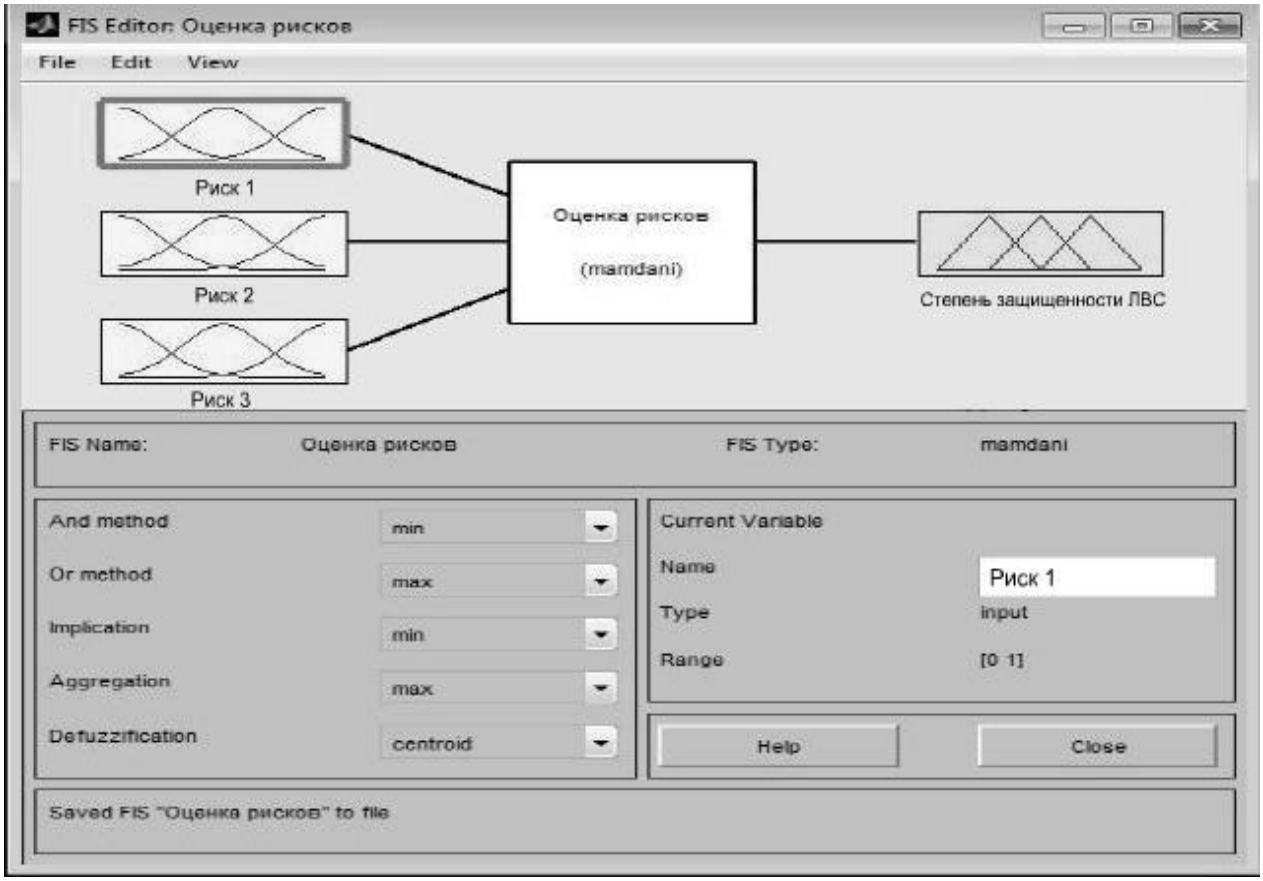


Рис. 3.2. Ввод входных переменных

Интерфейс позволяет пользователю построить график зависимости выходной переменной (степень защищенности сети), от любой из трех

входных переменных (группы рисков). В данном случае входными переменными являются три группы рисков, описанные выше. Применяя терминологию теории нечетких множеств, группы рисков это функции принадлежности отдельных термов системы нечеткого вывода. Функциям принадлежности был задан диапазон от 0 до 1, так как они выражают численную вероятность возникновения рисков соответствующей группы.

После редактирования функций принадлежности необходимо вызвать редактор правил нечеткого вывода и наиболее полно описать базу правил. База правил нечеткого вывода задается в виде логических выражений, например:

If «Риск 1» is «низкий» and «Риск 2» is «низкий» and «Риск 3» is «низкий» then «Степень защищенности» is «высокая».

Под выражением *«Риск 1 is «низкий»»* подразумевается, что вероятность возникновения угрозы, относящейся к первой группе рисков находится в диапазоне значений от 0 до 0,4 (значение средней вероятности от 0,4 до 0,6 , высокой от 0,6 до 1).

На основе базы правил с помощью алгоритма Мамдани (Mamdani) происходит фаззификация входных переменных. Процедура фаззификации устанавливает соответствие между численным значением входной переменной и значением функции принадлежности соответствующего ей терма лингвистической переменной. Следующей ступенью в алгоритме нечеткого вывода является агрегирование подусловий. На этом этапе происходит определение степени истинности каждого из утверждений, записанных в базе правил. Таким образом, если база правил изначально была задана логически верно, то выходная графическая модель будет более «плавной», без противоречивых точек на графике. Далее, по алгоритму происходит активизация подзаключений, представляющая собой вычисление произведения весового коэффициента и степени истинности нечеткого продукционного правила, а также аккумуляция заключений – вычисление функций принадлежности для каждой из выходных лингвистических переменных. Процедура дефаззификазции присваивает выходной лингвистической переменной конкретное численной значение [4].

На рис. 3.3 представлено окно вывода, где первые три столбца отображают значение соответствующих рисков, а последний степень защищенности сети. Вертикальными линиями в первых трех столбцах можно устанавливать необходимую величину риска, соответствующую значению для конкретной ЛВС.

Также с помощью функционала программы MATLAB можно отобразить трехмерный график зависимости выходной переменной от любых из двух входных переменных (рис. 3.4).

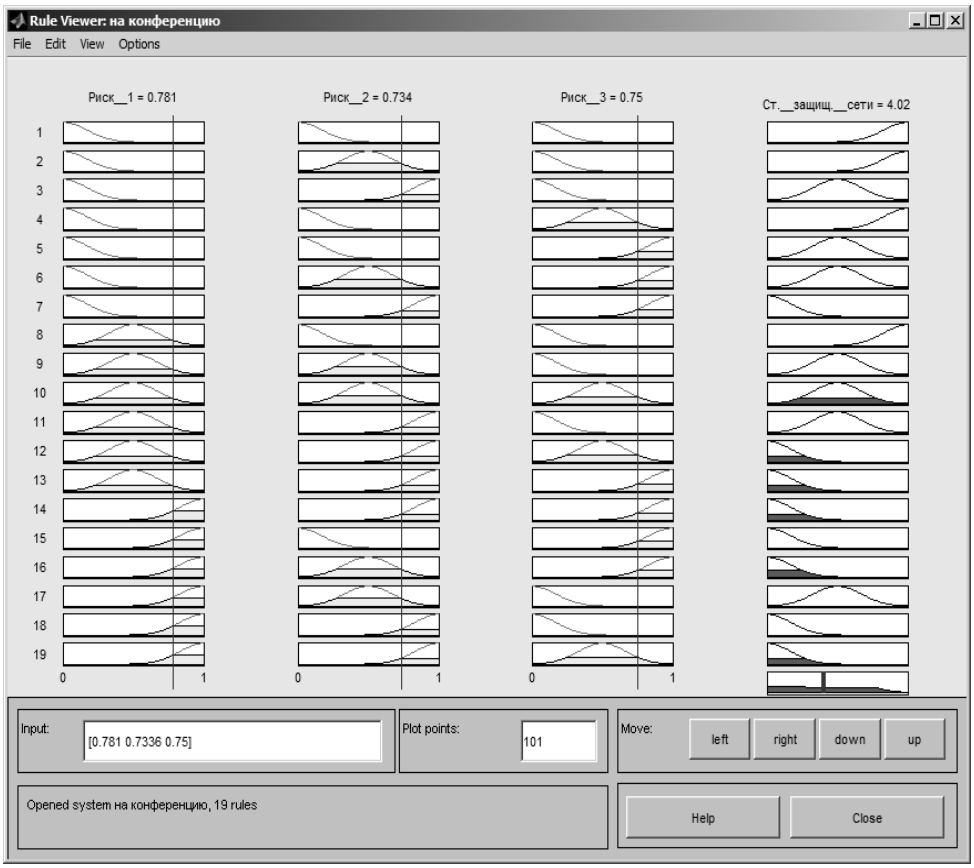


Рис. 3.3. Окно вывода

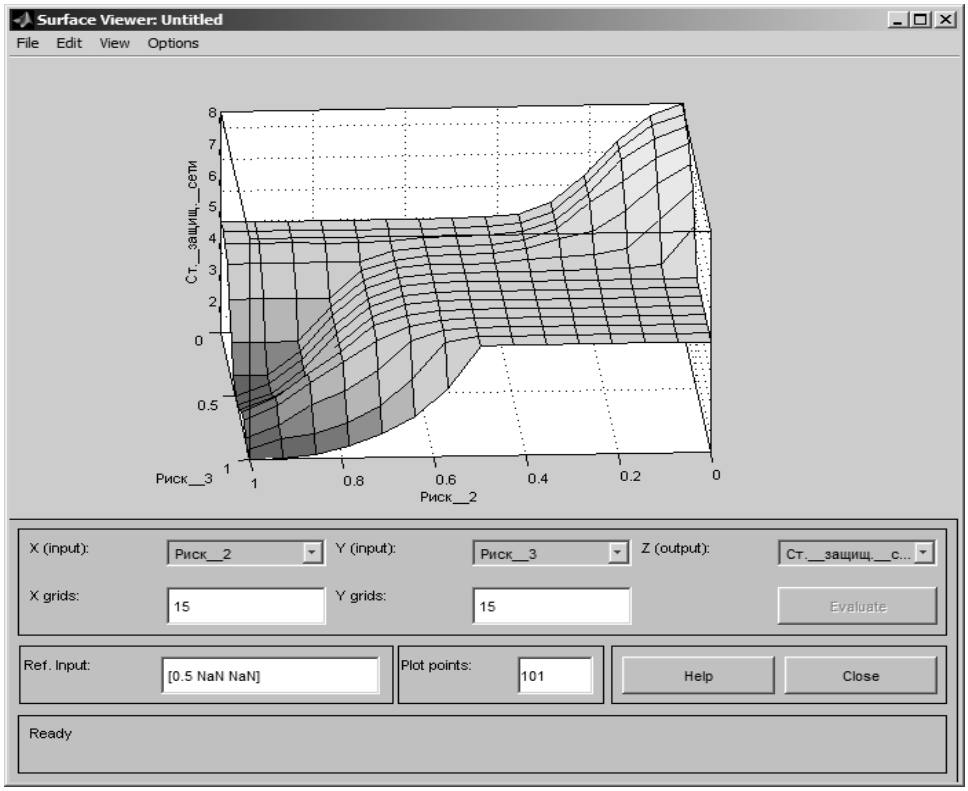


Рис. 1.4. Графическая модель

Построенная графическая модель отражает реальное состояние защищенности сети исходя из имеющихся показателей, что позволит выбрать более эффективные способы защиты сети и упростит систему управления рисками.

Литература:

1. Баранова Е.К., Гусев А.М. Методика анализа рисков информационной безопасности с использованием нечеткой логики на базе инструментария MATLAB // Образовательные ресурсы и технологии 2016. – №1 (13). – С.88-96. (3)
2. Глушенко С.А. Применение системы MATLAB для оценки рисков информационной безопасности организации // Бизнес-информатика 2013. – №4 (26). – С.35-42.
3. Национальный стандарт РФ ГОСТ Р ИСО/МЭК 27001–2006 Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. – М.: Стандартинформ, 2008. – 26 с.
4. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. – М.: Горячая линия-Телеком, 2007. – 288 с.

3.4. Исследование рисков хранения криптовалют на бирже

Мердина О.Д., Зельман С.Г.

Одним из основных вопросов, о котором следует задуматься при работе с криптовалютами – это вопрос о ее хранении. Как и обычные валюты, криптовалюты хранятся в кошельках. На сегодняшний день известно несколько существенно отличающихся друг от друга видов хранения криптовалюты на кошельках. Рассмотрим для начала виды кошельков криптовалют, их можно разделить по следующим параметрам:

- «холодный» и «горячий»;
- «тонкий» и “толстый»;
- бумажный;
- аппаратный.

«Холодный» кошелек – кошелек, который не имеет постоянного доступа к глобальной сети. То есть управление данным кошельком может осуществляться только при подключении его к глобальной сети.

«Горячий» кошелек постоянно подключен к сети. За счет этого он более уязвим. Именно такие кошельки используют криптобиржи.

«Толстый» кошелек – система, при которой на устройстве пользователя осуществляется хранение всех блоков цепи. Система ее полностью проверяет и контролирует.

«Тонкий» кошелек – кошелек, который осуществляет неполное хранение блокчейна. Такие кошельки хранят только данные пользователя. За счет этого данный способ хранения менее ресурсоемкий, но и менее безопасный.

Так же хранение криптовалют может осуществляться на бумажных носителях. Это специальный документ, на котором изображена копия ключа. Обычно счет выполнен в виде QR-кода.

Аппаратный кошелек – специальная аппаратная реализация хранения криптовалют, данный кошелек схож с «холодным» кошельком. Одним из его преимуществ является возможность хранение нескольких криптовалют [2].

Очень важно осуществлять хранение криптовалюты на тех кошельках, к которым у пользователя есть приватный ключ, который позволяет восстановить доступ к кошельку криптовалюты. И, конечно, этот ключ должен быть известен только данному пользователю.

События последнего времени заставляют задуматься о целесообразности хранения криптовалют на кошельках криптобирж. Взлом бирж, блокировка аккаунтов, вирусы, которые были найдены крупными корпорациями за последние годы показывают, что пользование биржами криптовалют требует особого внимательного подхода. Потеря средств на биржах криптовалют не регулируется законом и только при должном везении может случиться так, что утерянные или заблокированные средства будут возвращены их обладателю.

Сфокусируем свое внимание на проблеме взлома криптовалютных бирж, потому что от этого конечному пользователю сложно уберечься и невозможно защититься. Приведем конкретные примеры, иллюстрирующие риски хранения криптовалют на криптобиржах.

В августе 2016-го года произошел взлом третьей по объему операций криптовалютной биржи Bitfinex. Взломщикам удалось вывести около $65 млн, а стоимость криптовалют обвалилась примерно на 19% за несколько дней и на 5% за первые часы после взлома. Это был не первый случай взлома данной криптобиржи, так в мае 2015 года на бирже также была обнаружена уязвимость. Тогда злоумышленнику удалось вывести с биржи более 1500 BTC, что составляло всего лишь порядка 0,5% средств биржи [1].

В 2014 произошел похожий случай, тогда была взломана биржа Mt.Gox, но на тот момент она занимала первое место по объему операций. Тогда потери криптовалют составили эквивалент $450 млн. На тот момент взлом биржи обвалил цены криптовалют на 30%, а средства были безвозвратно утеряны пользователями криптобиржи и переведены на счета мошенников [4].

Взлом же 2016 года ставит больше вопросов в целесообразности использования криптобирж как места хранения криптовалюты. В отличии от биржи Mt.Gox, биржа Bitfinex имела двухфакторную авторизацию, которая осуществлялась с помощью самой биржи (1-ая стадия), а также компанией

BitGo, с которой данная биржа заключила договор. BitGo разработала решение, позволяющее пользователям быстрее проходить процедуру идентификации и вывода средств.

В данный момент подавляющее большинство бирж оставляет активными (доступными для проведения мгновенных операций) только небольшую часть активов. Остальные средства хранятся в пассивном режиме, то есть они недоступны для проведения операций.

Поначалу представители биржи Bitfinex объявили о том, что для взлома биржи нужно было преодолеть оба уровня аутентификации, чтобы получить доступ к средствам на бирже, но BitGo заявила, что их система не была взломана, что означает пробелы в самой бирже. Какая бы не была защита аккаунта на бирже, всегда есть риск пробелов в самой системе, от которой пользователю никак не защититься. В результате данного взлома биржи средства на ней были заморожены, а так как торговля на данной бирже проводилась не только в Bitcoin, но и в других криптовалютах, то пострадали все те пользователи, которые хранили средства на ней. Биржа прекратила проведение операций с криптовалютами на неопределенный срок.

Проверка пользователей, которые потеряли средства на бирже достаточно трудоемкая работа, при этом вероятность возврата средств пользователям, у которых украли средства была минимальной. Это означало для многих людей то, что их деньги повисли в воздухе, и они ничего с этим не смогли, бы сделать, до тех пор, пока биржу не разблокируют. Только через полгода, в конце января 2017-го года, представители биржи Bitfinex смогли найти неполадки в системе и сообщили о том, что за каждый возвращенный Bitcoin будет 5% вознаграждения, причем в любой той валюте, в которой пожелает владелец. Кроме того, взлом сильно повлиял на общую стоимость криптовалют.

На рис. 3.5-3.7 приведены курсы криптовалют Bitcoin и Monera после взлома криптовалютных бирж [3].

Во время взлома Bitfinex в мае 2015-го года украдено на тот момент было немного, но даже это повлияло на стоимость Bitcoin. Взлом Bitfinex в августе 2016-го года сильно обрушил стоимость валюты.

На рисунке 3.7 приведен курс криптовалюты Monero в кризисные для Bitcoin периоды. Как видно, он не сильно менялся.

Курс Monero показывает, что данная валюта может существовать независимо, в не малой степени из-за другого принципа его шифрования. Роль, стоимость и капитализация Monero резко увеличились в связи с тем, что требуется какая-то альтернатива Bitcoin’у, а после последних исследований выяснилось, что Bitcoin совершенно не такой анонимный в сравнении с Monero. Данный факт является ее неоспоримым преимуществом перед остальными криптовалютами.

Рис. 3.5. Курс Bitcoin во время взлома Bitfinex, май 2015-го года

Рис. 3.6. Курс Bitcoin во время взлома Bitfinex, август 2016-го года

Рис. 3.7. Курс Monero в исследуемый период времени с 2014 по 2017 годы

Исходя из вышеизложенного можно сделать достаточно простые выводы о безопасности хранения криптовалюты.

Во-первых, постоянное хранение средств в криптовалюте необходимо осуществлять на локальных кошельках.

Во-вторых, необходимо постоянно отслеживать информацию и отзывы о работе тех или иных криптобирж.

В-третьих, необходимо обратить внимание на бурно развивающиеся распределенные криптобиржи.

Дело в том, что когда криптовалюта находится на кошельке криптобирже, то есть большая вероятность, что эта криптобиржа будет взломана или прекратит свое существование, криптовалюта может быть потеряна. Именно для решения этой острой проблемы и были созданы распределенные криптобиржи, в которых активы находятся на личных кошельках и, таким образом, не могут быть потеряны при каких-либо проблемах криптобиржи. То есть проблемы, связанные с криптобиржами, никак не затронут ваши активы.

Для диверсификации рисков так же разумно использовать альтернативные криптовалюты, например, такие как Monero, потенциал которого до сих пор до конца не раскрыт.

Литература:

1. Биткоин-биржу Bitfinex взломали и вывели около $65 млн в эквиваленте // Geektimes [электронный ресурс]. URL: https://geektimes.ru/post/279148/ (дата обращения: 04.11.2017).
2. Виды кошельков для хранения биткоин и критерии выбора // MiningBitcoinGuide [электронный ресурс]. URL: https://miningbitcoinguide.com/kriptovalyuty/bitcoin/kak-vybrat-koshelek-btc#i-2 (дата обращения: 13.11.2017).
3. График биткоина к доллару // Coinspot [электронный ресурс]. URL: https://coinspot.io/charts/ (дата обращения: 04.11.2017).
4. Инсайд истории Mt. Gox // Хабрахабр [электронный ресурс]. URL: https://habrahabr.ru/post/214795/ (дата обращения: 04.11.2017).

3.5. Криптовалюты: риски сегодняшнего дня

Чернокнижный Г.М., Малахова П.А.

Увеличение числа виртуальных сообществ вызвано, с одной стороны, резким ростом технологических разработок, с другой – увеличением количества продвинутых пользователей Интернет, которые воспользовались новыми технологиями, в частности, для выполнения финансовых операций внутри сообществ. С этой целью создавались виртуальные валюты, часть из которых стала затем доступна для широкого круга пользователей. Основным преимуществом виртуальных валют является большая скорость платежных транзакций. К недостаткам следует отнести высокую волатильность обменного курса большинства криптовалют по отношению к традиционным валютам – а, следовательно, и к большинству товаров и услуг. К тому же, законодательная поддержка виртуальных валют значительно отстает от их технологического развития и, пока соглашения по данному вопросу не достигнуто, виртуальные валюты являются для преступников средством заработка и ценным финансовым инструментом одновременно. Они несут риски как способ мошенничества (отмывание денег, уклонение от уплаты налогов), но могут быть использованы и для более серьезных преступлений, например, финансирование терроризма.

Наиболее успешной виртуальной криптовалютой на сегодняшний день является биткоин [4]. Поэтому в плане информационной безопасности использования виртуальных денег будем рассматривать сеть Биткоин, т.к. она составляет примерно 3/4 общей рыночной стоимости всех виртуальных валют. Проблемы в альтернативных сетях – «альткоинах» – весьма схожи.

Эта валюта основана на пиринговой (одноранговой) сети и работает на глобальном уровне. Поэтому она может быть использована для всех видов транзакций (как для виртуальных, так и для реальных товаров и услуг). Биткоин делится до восьмого знака после запятой, позволяя использовать его для сделок самых различных размеров. Независимость биткоина от третьей стороны (он не контролируется государством, банками, никакими финансовыми учреждениями) предоставляет пользователям желаемый уровень конфиденциальности. По состоянию на 01 ноября 2017 года стоимость одного биткоина составляла 6,665 долларов США, однако курс может значительно меняться в течение одного дня [3]. Сейчас существует 863 криптовалюты, но, несмотря на это, Биткоин ни разу не терял свои лидирующие позиции на рынке виртуальных валют.

Биткоины создаются при помощи процесса, который называется «добыча» или майнинг (от англ. mining), являющимся по сути конкуренцией в процессе поиска решения математической задачи методом грубой силы. Каждый участник сети Биткоин имеет возможность быть майнером, используя

мощности своего компьютера для записи и проверки транзакций. Выпускается специальное аппаратное обеспечение для майнинга с большей эффективностью, чем стандартные аппаратные средства пользователя. Облачные услуги майнинга управляют так называемыми серверными фермами, сосредоточенными на майнинге. Они продают или сдают в аренду доли своих мощностей клиентам. Примерами таких организаций являются Hashflare и Genesis Mining. Протокол сети Биткоин имеет встроенные алгоритмы, регулирующие функцию майнинга по сети, а также ограничивает общее количество монет, которые будут выпущены. Ограничение является фиксированной суммой в 21 миллион биткоинов.

В основу многих виртуальных валют, в том числе, биткоина, положена технология блокчейн, децентрализованного метода слежения за транзакциями в сети. Здесь используются механизмы цифровой подписи, хэш-цепочки (функции хеширования SHA256, SHA512, RIPEMD160, Scrypt и др.) и схема доказательства работы (Proof of Work). Сложность вычислений увеличивается прямо пропорционально увеличению вычислительной мощности в сети. Как альтернатива данной модели была разработана proof-of-stake (PoS) или «Доказательство доли». Она учитывает количество расчетных единиц виртуальной валюты, принадлежащих каждому пользователю сети. Использование PoS дает возможность устранить некоторые уязвимости PoW, такие как высокое потребление энергии и атаку 51%. В PoW-системе лицо, контролирующее вычислительную мощность равную или выше 51% от мощности всей сети, способно управлять функционированием системы и манипулировать транзакциями, например, осуществлять двойную трату средств или блокировать подтверждение определенной транзакции. В PoS-системе для аналогичной атаки лицу должно принадлежать 51% всех расчетных единиц в сети, что гораздо дороже и сложнее, чем получить контроль на 51% вычислительных мощностей.

Виртуальная монета, по определению Сатоши Накамото, это цепочка цифровых подписей. Каждый владелец валюты имеет пару ключей: публичный, (который и является адресом кошелька) и закрытый. Ключи предоставляют право на совершение транзакций в сети, на расшифровывание полученных значений для их передачи другому пользователю. Эти ключи хранятся в файле на компьютере пользователя и, если этот файл будет утрачен, то все биткоины, связанные с ключом будут также утрачены. Эти ключи могут также храниться на любых физических носителях или в облаке. Важно помнить, что кошелек также не содержит единиц валюты в буквальном смысле, а только предоставляет доступ к средствам на счете, которые хранятся в блокчейне.

Схема транзации биткоина представлена на рис. 3.8. Для начала транзакции будущий владелец P1 должен отправить свой публичный ключ пер-

воначальному владельцу P0. Этот владелец передает биткоины, подписывая хэш предыдущей транзакции и публичный ключ будущего владельца своим закрытым ключом.

Рис. 3.8. Схема транзакции биткоина

Каждый биткоин несет в себе всю историю транзакций, через которые он прошел, и любая передача от одного пользователя к другому становится частью кода. Биткоин хранится таким образом, чтобы новый владелец был единственным человеком, способным его потратить. Все подписанные транзакции, отправляющиеся в сеть, являются открытыми для публики, однако не дается никакой информации об участниках сделки. В отсутствии третьей доверенной стороны – посредника, способного подтвердить транзакцию, стоит вопрос, как избежать двойную трату средств, если монета оказалась скопированной или подделанной. Решение данного вопроса лежит в идее «временной метки», онлайн-механизме, который гарантирует, что данные не менялись с определенного момента времени для того, чтобы попасть в хэш. Каждая временная метка включает в себя предыдущую метку в своем хэше, формируя цепочку владения. Транслируя новые транзакции, сеть может их подтверждать при помощи майнеров, которые предоставляют вычислительные мощности для подтверждения транзакций.

Не обсуждая здесь вопросы лицензирования и перспективы развития криптовалют на долгие годы, рассмотрим угрозы и риски в их использовании, которые существуют на сегодняшний день.

Модель безопасности сети Биткоин основывается на независимости подтверждения транзакций майнерами и на децентрализованном контроле над ключами. То есть для обеспечения безопасности сети Биткоин, нужно просто оставаться в рамках модели безопасности этой сети – секретные ключи пользователей должны находиться только под их контролем и транзакции не должны проводиться вне блокчейна [2].

Риски использования виртуальной валюты при выполнении транзакций

1. *Риск потери биткоинов*. Следует помнить, что сеть Биткоин является сетью с открытым кодом, что подразумевает под собой возможность создания программного обеспечения со своими условиями любым пользователем сети. Данная возможность увеличивает вероятность человеческой ошибки, которая может привести к искажению адресов входов и выходов транзакций, что является угрозой потери биткоинов. Под потерей биткоинов подразумевается их безвозвратная потеря из сети Биткоин вообще, а не только с пользовательского счета.

Аналогичный риск потери биткоинов возникает, когда пользователь теряет свой секретный ключ. Тогда все средства на публичном ключе, который соответствует утерянному приватному, пропадают навсегда. Так или иначе, не все пользователи создают резервные копии ключей, и далеко не все жесткие диски могут быть долговечными. Эти факторы всегда могут стать причинами потери ключей, а значит и самих биткоинов, а значит привести к дефляции валюты.

2. *Риск угрозы двойной траты*. Майнеры в сети находятся в постоянной вычислительной гонке по решению блоков для того, чтобы их цепочка блоков была записана первой, и за каждый блок была получена награда. Существует вероятность того, что злоумышленником будет создана более длинная цепочка блоков, которая будет содержать аналогичную транзакцию из цепочки блоков майнера, но будет являться фальсифицированной (например, с выходом на адрес кошелька злоумышленника). Тогда в блокчейн будет записана цепочка блоков злоумышленника как подтвержденная, а подлинная транзакция отправится в массив неподтвержденных транзакций, и со временем узлы сети удалят ее как ложную. То есть произойдет двойная трата одной транзакции.

Чтобы реализовать данную атаку, злоумышленнику требуется заранее решить цепочку блоков и транслировать ее в сеть Биткоин в нужный момент. К счастью, сложность решения математических задач в каждом из блоков делают данную атаку в этом виде невыполнимой. Хэш-результат решения каждого блока должен быть меньше определенного значения (цели сложности сети), для этого в конец блока методом грубой силы подставляются случайные значения. Когда подобранная хэш-сумма соответствует цели сложности сети, блок считается решенным, и эта хэш-сумма становится уникальным маркером этого блока.

При изменении хотя бы одного символа в блоке, его хэш-сумма изменяется кардинально. А так как хэш-сумма одного блока содержит в себе и ссылку на предыдущий блок, следовательно, ни один из блоков не может быть подменен, так как хэш-сумма подмененного блока будет другой.

В результате можно сделать вывод, что невозможно заранее просчитать ветку блоков, так как злоумышленнику требуется знать хэш-сумму того блока, с которого он хочет начать. Даже если в его распоряжении будут тысячи компьютеров, не удастся решить цепочку блоков раньше, чем это сделают другие пользователи сети по причине того, что злоумышленнику предстоит соревноваться не с несколькими майнерами, а со всей сетью Биткоин.

Для успешной реализации угрозы двойной траты злоумышленнику потребуется больше половины мощности всей сети Биткоин. Тогда появляется вероятность 50%, с которой он может решить хотя бы один блок первым.

В блокчейне блоки строятся последовательно, поэтому более ранние транзакции надежнее новых. Так что система уязвима к двойной трате только на конце цепочки блоков, следовательно, рекомендуется подождать несколько блоков, прежде чем считать транзакцию подтвержденной.

3. *Риск необработки транзакций*. Как говорилось ранее, всего будет выпущено 21 млн. биткоинов. Когда остановится выпуск новых монет, у майнеров пропадет смысл в решении блоков, так как награды за их решение больше не будет. Но следует помнить, что к любой транзакции можно добавить комиссию (fee), то есть плату за ее обработку. Майнер получает эту комиссию вдобавок к награде за блок при обработке такой транзакции. На данный момент в сети обрабатываются все транзакции, так как главной мотивацией майнеров является награда за блок. Но в ближайшем будущем транзакции будут обрабатываться в зависимости от комиссий, которые в них включены. Транзакции без комиссий вероятно будут игнорированы. Так что имеет место риск необработки транзакций без комиссий. Тем не менее, эти комиссии скорее будут намного дешевле банковских.

4. *Риск потери доверия сети*. В настоящее время один обычный компьютер может потратить на решение одного блока несколько лет, так что в этом случае вероятность решения блока раньше остальной сети (примерно 10 минут) близка к нулю. Стоимость специального оборудования (чипов, видеокарт и т.д.) в таком количестве, чтобы работать в сети в одиночку, достаточно высокая. Для получения более стабильного заработка, майнеры вступают майнинг- пулы (mining pool), где решение блоков происходит коллективно, а вознаграждение распределяется пропорционально проделанной работе. Некоторые майнинг-пулы достигают свыше 20% мощности всех узлов сети. Этот факт, что некоторые пулы достаточно большие, порождают риски безопасности.

Как уже упоминалось, у злоумышленника очень мало шансов решить цепочку блоков раньше остальной сети, но это все-таки возможно. Вероятность растет с ростом вычислительной мощности злоумышленника по

сравнению с остальной сетью. Например, такой пул, как BTC Guild (Гильдия BTC) самостоятельно решила цепочку из шести блоков, впоследствии добровольно ограничив членство в пуле, во избежание недоверия сети Биткоин [3].

Сейчас при проведении небольших транзакций, следует дождаться углубления транзакции в блокчейн хотя бы на 1 блок (одно подтверждение). При проведении крупных транзакций, желательно подождать шесть подтверждений, то есть 6 блоков.

Практическая безопасность пользователей

1. *Хранение биткоинов на физических носителях*. Большинство пользователей гораздо лучше знакомы с физической безопасностью, чем с информационной. Поэтому одним из эффективных способов защиты биткоинов является их преобразование в физическую форму. Все секретные ключи – это длинные числа, которые могут храниться, например, на бумаге. Тогда защита средств заключается в физической безопасности печатной копии ключей Биткоина. Такое хранение «вне сети» называется холодным хранением. В виду отсутствия доступа в Интернет, данный способ является наиболее эффективным.

2. *Аппаратные кошельки*. В отличие от смартфонов и компьютеров, которые имеют доступ в Интернет, у аппаратного кошелька для биткоинов единственная функция – надежное хранение биткоинов. Аппаратные кошельки обеспечивают высокий уровень безопасности, так как в них отсутствует какое-либо лишнее ПО, и их интерфейс достаточно ограничен. На сегодняшний день существует много таких кошельков, но самыми популярными из них считаются Ledger Nano S, KeepKey и Trezor. В дальнейшем, аппаратная форма обеспечения защиты от внешних воздействий будет принимать все большую популярность для обеспечения безопасности биткоинов.

3. *Взвешивание рисков*. Каждый пользователь сети Биткоин боится потерять свои средства, а, соответственно, и кражи своих биткоинов с кошелька. Но не стоит забывать, что биткоины могут быть не только украдены, но еще и потеряны, что имеет даже большую вероятность случиться. При попытке обезопасить свои цифровые кошельки и приватные ключи, пользователи создают сложную цепочку зашифрованных резервных копий, ключи от которых в конечном итоге также могут быть утеряны. В этом случае резервные копии перестают иметь смысл, а значит риск потери биткоинов становится даже выше, чем если бы этих резервных копий не было совсем.

4. *Диверсификация активов*. Для обычного пользователя хранение биткоинов централизованно на одном кошельке является самым удобным способом. Но для обеспечения достаточной безопасности своих средств,

также можно распределить биткоины между несколькими цифровыми кошельками, аппаратными кошельками и физическими носителями.

Расчет вероятности осуществления двойной траты

Рассмотрим ситуацию осуществления двойной траты злоумышленником. Допустим, злоумышленнику удается сгенерировать альтернативную цепочку блоков раньше, чем вся сеть решит подтвержденную цепочку блоков. Осуществление данной атаки не позволет злоумышленнику в дальнейшем делать какие-либо произвольные изменения в сети, например, создавать ценности из воздуха или получать средства, которые никогда ему не принадлежали. Пользователи не примут альтернативную транзакцию в качестве оплаты, а майнеры не будут принимать блок, содержащий эти транзакции.

Сатоши Накамото была просчитана вероятность осуществления двойной траты в своем докладе в 2008 году [4]. Исходя из его расчетов, можно сказать, что эта вероятность экспоненциально падает с ростом числа блоков. Результаты расчетов представлены в таблицах 3.3 и 3.4, где p – вероятность решения следующего блока подтвержденным узлом, q – вероятность решения следующего блока злоумышленником, z – количество решенных блоков, P – вероятность обгона злоумышленником новых участников.

Таблица 3.3

Результаты расчета вероятности осуществления двойной траты относительно q [4]

$q = 0.1$		$q = 0.3$	
$z = 0$	$P = 1.0000000$	$z = 0$	$P = 1.0000000$
$z = 1$	$P = 0.2045873$	$z = 5$	$P = 0.1773523$
$z = 2$	$P = 0.0509779$	$z = 10$	$P = 0.0416605$
$z = 3$	$P = 0.0131722$	$z = 15$	$P = 0.0101008$
$z = 4$	$P = 0.0034552$	$z = 20$	$P = 0.0024804$
$z = 5$	$P = 0.0009137$	$z = 25$	$P = 0.0006132$
$z = 6$	$P = 0.0002428$	$z = 30$	$P = 0.0001522$
$z = 7$	$P = 0.0000647$	$z = 35$	$P = 0.0000379$
$z = 8$	$P = 0.0000173$	$z = 40$	$P = 0.0000095$
$z = 9$	$P = 0.0000046$	$z = 45$	$P = 0.0000024$
$z = 10$	$P = 0.0000012$	$z = 50$	$P = 0.0000006$

Предположим, что $p > q$, тогда вероятность падает экспоненциально, так как количество блоков, которые злоумышленник должен догнать, увеличивается. Если злоумышленнику не удастся обогнать решение блоков на первых этапах, то его шансы невероятно быстро стремятся к нулю.

Таблица 3.4

Результаты расчета параметры атаки двойной траты относительно *P* [4]

$P < 0.001$	
$z = 5$	$q = 0.10$
$z = 8$	$q = 0.15$
$z = 11$	$q = 0.20$
$z = 15$	$q = 0.25$
$z = 24$	$q = 0.30$
$z = 41$	$q = 0.35$
$z = 89$	$q = 0.40$
$z = 340$	$q = 0.45$

Рекомендации по уменьшению экономических рисков

Виртуальные валюты на данном этапе не представляют большого риска для экономики. Пока еще нет такого государства, которое бы приравняло виртуальную валюту к официальному платежному средству. Поэтому на настоящий момент виртуальные валюты существуют за пределами банковской системы, то есть отсутствует их связь с реальной экономикой. На стабильность виртуальных валют влияют количество пользователей, их доверие и возможные уязвимости в криптоалгоритмах. Относительно фиатных валют, виртуальные валюты обладают значительной неустойчивостью, поэтому их ввод в реальную экономику может привести к нестабильности в банковской системе.

Для того чтобы стабилизировать децентрализованную виртуальную валюту при ее вводе в реальную экономику, можно оказать влияние на сложность сети. Для этого требуется оказать влияние на мощность сети, то есть повлиять на мощность каждого пользователя. Чтобы это осуществить, предлагается зарегистрировать всех майнеров и регламентировать их максимальную мощность на международном уровне, а также установить сумму комиссии за обработку транзакций. Оказав влияние на сложность сети, можно стабилизировать виртуальную валюту, а значит, минимизировать потенциальные экономические риски, при введении виртуальной валюты в банковскую систему.

Приведенные результаты позволяют сделать вывод, что существующая модель угроз для обмена виртуальными деньгами в глобальной сети, как и модель нарушителя, для случая использования криптовалют не вполне адекватно отвечает ситуации сегодняшнего дня. Поэтому здесь необходимо создание формальных моделей, методов и средств оценки информационной безопасности процесса вхождения криптовалют в состав официальных платежных систем. В первую очередь, следует привлекать для этой цели специалистов по криптографической защите информации [1].

Литература:

1. Васильева И.Н. Криптографические методы защиты информации. Учебник и практикум для академического бакалавриата. – М.: Изд-во Юрайт, 2016. – 349 с.
2. Andreas M. Antonopoulos. Mastering Bitcoin. – Sebastopol: O'Reilly Media, Inc., 2017. – 136 p.
3. Bits media BTC Guild. Русскоязычный информационный сайт о криптовалюте Bitcoin [сайт]. URL: https://bits.media (дата обращения 01.11.2017).
4. Nakamoto Satoshi. Bitcoin // Bitcoin Project [электронный ресурс]. URL: https://bitcoin.org/bitcoin.pdf (дата обращения 01.11.2017).

3.6. Риски электронного банкинга

Соловьев А.И., Соловьев С.А.

Проблемы применения интернет-технологий в банковской деятельности

Современный банковский бизнес активно внедряет современные достижения в сфере информационно-коммуникационных технологий (ИКТ). Они легли в основу развития такого вида банковских услуг как дистанционное банковское обслуживание (ДБО). Это позволяет снижать себестоимость предоставления банковских услуг и повышать конкурентоспособность кредитной организации.

Внедрение интернет технологий является одним из наиболее эффективных способов модернизации инфраструктуры в экономике.

Интернет-технологии становятся все активнее внедряются в политику, государственное управления, бизнес, трансформируют характер производственных в экономике и межличностных отношений в обществе. В принятой в июле 2017 года программе «Цифровая экономика в Российской Федерации»[5] определены цели, задачи, направления и сроки реализации основных мер государственной политики по созданию необходимых условий для развития в России цифровой экономики, в которой «данные в цифровом виде являются ключевым фактором производства во всех сферах социально-экономической деятельности»[6].

К отличительным особенностям, оказывающим влияние на социально-экономические процессы следует отнести [1]:

---

[5] «Цифровая экономика в Российской Федерации». Программа утверждена Распоряжением от 28 июля 2017 года №1632-р. – URL: http://government.ru/docs/28653/

[6] «Цифровая экономика в Российской Федерации». – URL: http://government.ru/docs/28653/

− интернет способствует развитию глобальной конкуренции компаний и организаций, в том числе, кредитных организаций, независимо от места их расположения;

− глобальная конкуренция способствует внедрению международных стандартов и соответствия им;

− тенденции активного использования ИКТ распространяются и на банковскую сферу.

Использование Интернета порождает многие новые виды рисков, которые базируются на использовании ИКТ.

Риски интернет-банкинга

Понятие дистанционного банковского обслуживания, реализуемого с помощью «интернет-банкинг», включает в себя возможность предоставления клиентам банков к своим счетам и совершения операций по ним, к информации о банковских услугах и видам обслуживания с использованием компьютерных технологий. Для частных клиентов это реализуется в системе «интернет-банка» (on-line банка), а корпоративные клиенты – в системе «банк-клиент».

К числу доступных операций относятся, например, управление наличностью, осуществление электронных переводов и платежей, автоматизация клиринговых операций, предъявление счетов к оплате, межбанковские переводы средств, депозитные и кредитные операции, инвестиционные решения, получение информации о транзакциях, состоянии баланса и других видов деятельности, приносящей доход.

Банки начали опытную эксплуатацию различных форм оперативного удаленного банковского обслуживания (online banking) и проводили эксперименты в течение многих лет. Первоначально эти эксперименты проводились с закрытыми системами, в которых клиенты получали доступ к банку через посредство телефонного набора или кабельного соединения. Эти системы ограничивали потенциальную клиентскую базу банка. При развитии и расширении сети Интернет клиенты получили возможность воспользоваться этой технологией в любом месте мира для того, чтобы получить доступ к сети связи своего банка. Интернет сделал банковские услуги и обслуживание доступными для большого, практически неограниченного, числа клиентов и устранил барьеры, обусловленные географическими факторами и правами собственности на системы.

Снижение количественных показателей, включая конкурентные затраты на обслуживание клиентов, стали стимулировать банки к переоценке значения ИКТ и пересмотру своих стратегий в части электронной коммерции и интернет-банкинга. Это способствовало быстрому увеличению числа клиентов, использующих банковские услуги и получающих обслуживание в режиме on-line. Сложность ситуации для банков заключается в том, чтобы

убедиться, что выгоды от применения технологии интернет-банкинга превышают потери из-за затрат и рисков, связанных с ведением бизнеса в киберпространстве.

Чтобы оценить риск, банковским менеджерам необходимо понимание стратегий и используемых технологий, чему может способствовать сопоставления аналогичных показателей различных банков. В число ряда рыночных факторов, которые определять стратегию развития ДБО в банке, входят следующие:

- конкуренция;
- эффективность затрат;
- географический охват;
- торговая марка.
- демография клиентуры.

Оценки менеджерами банков рисков, сопутствующих расширению ИКТ в банковской сфере, способствует понимание сути различных типов услуг интернет-банкинга [2]. В настоящее время получили распространение три основных варианта интернет-банкинга:

Информационный, как базовый уровень интернет-банкинга. Как правило, банк при этом дает информацию о банковских услугах на обособленном сервере. Соответствующий риск при этом низок, поскольку информационные системы при этом не имеют непосредственной связи между таким сервером и внутренней вычислительной сетью банка. Хотя риск для банка сравнительно невелик, соответствующий сервер или web-сайт может оказаться уязвимым для злонамеренных воздействий.

Коммуникационный – это тип системы интернет-банкинга, позволяющий реализовать отдельные виды взаимодействия между системами конкретного банка и его клиентом. Такое взаимодействие ограничивается электронной почтой, запросами справок о счетах, заявками на ссуды или внесение изменений в стандартные файлы (изменение имени и адреса). В этом варианте иметь место связь с внутренними вычислительными ресурсами банка, что повышает сопутствующий риск выше, чем в случае чисто информационного варианта системы. Возрастают требования к средствам контроля для мониторинга, предотвращения и оповещения руководства о возможных попытках неавторизованного доступа к внутренним сетям и компьютерным системам банка. Становится существенно более важным осуществление противовирусного контроля.

Операционный – уровень интернет-банкинга, предоставляющий клиентам возможности проведения банковских транзакций. Транзакции клиента включают доступ к счетам, осуществление платежей, перевод средств и т. п. Это обеспечивает непосредственная связь сервера Интернет сети и внутренней вычислительной сети банка. Такой архитектуре сопутствует

наивысший риск и в ней должны быть обеспечены самые серьезные средства контроля.

Рассмотрим виды рисков, которые сопутствую внедрению ДБО с использованием ИКТ.

Интернет-банкинг создает для банков ряд проблем, связанных с контролем над рисками [3]. Риск, как правило, связывают с ситуациями, когда ожидаемые или непредвиденные обстоятельства могут оказать негативное влияние на доходы или капитал банка. В США, например, в интересах банковского надзора ОСС[7] определило девять категорий риска. Этими рисками отнесены: кредитный, процентный, риск ликвидности, ценовой, валютный, операционный, риск несоответствия, стратегический и репутационный риски. Перечень не может считаться исчерпывающими, но все перечисленные риски связывают с интернет-банкингом.

Кредитный риск – риск для доходов или капитала банка, возникающий из-за неспособности лица, принявшего на себя обязательства (должника), выполнить требования по кредитному договору, заключенному с банком.

Интернет-банкинг предоставляет банку возможность расширения географии своей работы и предоставления клиентам получать банковское обслуживание, в том числе предоставление кредита данным учреждением, практически из любого места в мире. Взаимодействие с клиентами через Интернет, когда отсутствует какой-либо личный контакт, порождает для банка проблему верификации (подтверждения) истинности личностей клиентов, что является важным элементом при принятии правильных решений в части кредитования. Могут оказаться проблемными вопросы подтверждения залога и/или иного обеспечения безопасности кредита в случаях работы с удаленными заемщиками. Отсутствие правильного управления интернет-банкингом может способствовать концентрации кредитов заемщиков, не верифицированных должным образом, или кредитов в отдельной отрасли производства. Нельзя оставить без внимания вопрос того под контролем какой юрисдикции находятся взаимоотношения порождаемые через Интернет.

Процентный риск – риск для доходов или капитала, возникающий из-за движения процентных ставок. Банк концентрирует свое внимание на чувствительности размера своих активов, обязательств и доходных статей к изменениям процентных ставок. Процентный риск может возникать из-за различий между моментами изменения ставок и моментами движения средств (риск переоценки), из-за меняющихся соотношений процентных ставок между разными кривыми доходности активов банка (базовый риск),

[7] Управление контролера денежного обращения США (ОСС)

а также из-за вариантов процентного дохода, заложенных в услугах банка (опционный риск).

Интернет-банкинг способствует формированию депозитных, кредитных и иных отношений с широким кругом потенциальных клиентов. Расширенный доступ к клиентам, заинтересованным в наиболее высоких процентных ставках или сроках, должен сопровождаться усилением контроля руководства банка и поддержания систем управления активами/пассивами, включая способность быстрого реагирования на меняющиеся условия формирования банковских активов и пассивов.

Риск ликвидности – риск для доходов или капитала, обусловленный неспособностью банка выполнять в срок свои обязательства без неприемлемых для него потерь. Риск ликвидности включает неспособность управлять внеплановыми изменениями в источниках финансирования. Риск ликвидности может возникать из-за ошибок в оценке изменений рыночных условий, влияющих на способность банка быстро реализовать активы с минимальными потерями в их стоимости.

Интернет-банкинг способствует росту изменчивости в депозитах, движении средств, поступающих от клиентов, которые открывают свои счета только из соображений ставок или сроков. Системы управления активами/пассивами и кредитным портфелем должны соответствовать услугам, предлагаемым в рамках интернет-банкинга. Усиленный мониторинг ликвидности и изменений в депозитах и ссудах должен быть оправдан с учетов объема и характера действий клиентов со счетами через Интернет.

Ценовой риск – риск для доходов или капитала, возникающий из-за изменений в стоимости портфелей финансовых инструментов. Этот риск появляется из рыночных предположений, сделок и позиций, занимаемых на рынках процентных ставок, акций и других биржевых активов.

Банки подвержены ценовому риску, если они расширяют депозитный брокеринг, торговлю кредитами или программы страхования от риска в результате деятельности в рамках интернет-банкинга. Осуществление активного трейдинга должно быть обеспечено наличием необходимых систем управления для мониторинга, измерения ценового риска и управления им.

Валютный риск возникает когда кредитный портфель деноминируется в валюте или финансируется за счет займов в иностранной валюте. В ряде случаев банки вовлекаются в мультивалютные кредитные обязательства, позволяя заемщикам выбирать ту валюту, которую они предпочитают. Валютный риск увеличивает действие политических, социальных или экономических факторов. Соответствующие последствия могут оказаться неблагоприятными, если при обмене валют наблюдаются сильные колебания ее обменного курса.

Операционный риск является постоянным риском для доходов и капитала банка, обусловленным мошенничеством, ошибками и сбоями при предоставлении банковского обслуживания, поддержания конкурентной позиции и управления информацией. Он проявляется в каждой услуге и виде обслуживания, предлагаемых банком, и охватывает все процессы предоставления услуг и видов обслуживания клиентов услуг, обработку транзакций. Он должен учитываться при разработке систем, в частности, компьютерные системы. Он увеличивается с ростом сложности услуг и обслуживания и зависит от условий осуществления внутреннего контроля.

Внедрение интернет-банкинга влечет высокий уровень операционного риска, особенно если ведение бизнеса неадекватно спланировано, реализовано и недостаточно контролируется. Банки, предоставляющие услуги и обслуживание через Интернет, стремятся соответствовать ожиданиям своих клиентов для формирования высокого уровня доверия к своей торговой марке. Клиенты ожидают непрерывной доступности к услугам и сервисам на сайте банка с простой навигацией по ним. Осуществляя деловые операции через Интернет, они должны быть защищены от ошибок или промахов со стороны финансовых учреждений, которые не обладают специализированными средствами внутреннего контроля для управления проведением операций в рамках интернет-банкинга.

Основного внимания заслуживают риски атак или попыток проникновения в банковские компьютерные и сетевые системы. Банкам следует иметь надежные средства защиты и обнаружения, чтобы обезопасить свои системы интернет-банкинга от вторжений как изнутри, так и снаружи. Важную роль играют вопросы правильной организации резервирования как данных, так и аппаратных средств, чтобы при выходе из строя основного сервера происходила автоматическая переадресация потока данных на резервный сервер, размещенный в другом месте. При этом следует уделять особое внимание вопросам информационной безопасности. В ситуациях такого рода средства обеспечения безопасности и внутреннего контроля на резервных каналах должны быть такими же по сложности, как и те, которые имеются на основном месте обработки.

В дополнение к операционному риску ошибки и сбои в предоставлении ведут к усугублению репутационного риска, риск ликвидности и кредитного риска.

Риск несоответствия – риск для доходов или капитала, возникающий из-за нарушений законов, правил, инструкций, предписанной практики или этических стандартов либо не соответствующих им действий. Риск несоответствия подвергает учреждение опасности штрафов, административных денежных взысканий, компенсации ущерба и нарушения контрактов. Риск несоответствия ведет к ухудшению репутации, сокращению

льгот, ограничению деловых возможностей, снижению возможностей расширения, а также неполному исполнению договоров. Соответственно, банкам необходимо следить за тем, что информация и услуги, которые они предоставляют по каналам интернет-банкинга, соответствуют современному уровню требований, правил и положений регламентирующих их деятельность[8].

Стратегический риск отражает перспективное влияние на доходы или капитал ошибочных управленческих решений несоответствия их исполнения, в том числе, недостаточная способность реагировать на изменения в отрасли. Уровень риска зависит от совместимости стратегических целей банка, его деловой стратегии, разработанной для достижения этих целей, ресурсов, отведенных для их реализации, и качества реализации стратегий в целом. Ресурсы, требуемые для осуществления деловых стратегий, включают, в числе прочих, организацию каналов связи, используемые операционные системы, сети доведения услуг, а также средства и системы управления ими с учетом воздействия экономических, технологических, конкурентных, регулятивных и других факторов.

Руководство банка должно осознать риски, связанные с интернет-банкингом, до принятия решения о разработке и внедрения конкретного вида услуг, предоставляемых посредством интернет-банкинга. Для поддержки такого рода решений необходимы высокий уровень развития технологий и информационных систем управления в банке.

Новые технологии, особенно Интернет, способны вносить быстрые изменения в конкуренцию. Поэтому банковские эксперты в области технологий совместно с маркетинговым и операционным персоналом должны участвовать в процессе планирования и принятия решений, чтобы не выйти за пределы устойчивости банка к риску.

Репутационный риск представляет собой текущее и перспективное влияние на доходы и капитал банка, которое может оказать изменение в общественном мнении. От него зависит способность банка продолжать обслуживать существующую клиентуру и устанавливать взаимоотношения с новыми клиентами. Потеря репутации, как правило, сопровождается судебными преследованиями учреждения, финансовым потерям в следствие сокращения или утраты клиентской базы. Репутационный риск пронизывает

---

[8] Одним из первых документов Центрального банка Российской Федерации по тематике интернет-банкинга было Письмо Банка России от 3 февраля 2004 г. № 16-Т «О Рекомендациях по информационному содержанию и организации web-сайтов кредитных организаций». В настоящее время действует обновленная версия данного документа - Письмо Банка России от 23 октября 2009 г. № 128-Т «О Рекомендациях по информационному содержанию и организации Web-сайтов кредитных организаций в сети Интернет».

всю деятельностью банка, в том числе, в сфере интернет-банкинга, который становится «лицом банка» и по его организации складывается мнение о банке в целом.

Репутация банка падает, если он не способен соответствовать требованиям рынка и обеспечить точное, своевременное обслуживание. Ей может быть нанесен ущерб при обслуживании в рамках интернет-банкинга, при его плохой организации, отталкивающей клиентов и общественность. Хорошо организованный сайт банка, включающий раскрытие информации, т.е. реализующий «политику открытости», может являться одним из способов ограничить репутационный риск. Клиенты должны понять, что они могут обоснованно ожидать от той или иной банковской услуги, а также каким рискам они могут подвергаться и какие выгоды они получают в случае использования данной системы.

Банкам необходимо быть уверенными в том, что их планы по обеспечению непрерывности деловых операций в полной мере реализуются в рамках интернет-банкинга. Регулярная проверка плана по обеспечению непрерывности операций, включая стратегии взаимодействия с прессой и общественностью, поможет банку гарантировать, что он способен эффективно и должным образом реагировать на любые негативные проявления к своей репутации со стороны клиентов и средств массовой информации.

Представленная классификация банковских рисков, имеющих непосредственную связь с организацией интернет-банкинга не может рассматриваться как окончательный вариант. Все большее проникновение в нашу жизнь ИКТ, в том числе в банковскую сферу, создают не только впечатляющие возможности развития бизнеса, но и сопровождаются изменением «ландшафта рисков», порождаемых этими же возможностями. А это, в свою очередь, формирует задачу совершенствования систем управления рисками, в том числе, при использовании интернет-банкинга.

Литература:

1. Соловьев А.И. Современные проблемы безопасности электронной коммерции. // Экономико-организационные и программно-технические вопросы обработки и защиты информации. - СПБ: СПб ГИЭУ, 2003. – С/48-32.

2. Горшков В.В., Соловьев А.И. Экономическая безопасность электронной коммерции: современное состояние и перспективы. // Актуальные проблемы финансов и банковского дела: Сб. научн. тр. Вып. 6 / под ред. А.И. Михайлушкина, Н.А. Савинской. – СПб: СПбГИЭУ, 2003. – С.138-141.

3. Сычев А.М., Ревенков П.В., Дудка А.Б. Безопасность электронного банкинга. – М.: РФК-Имидж ЛАБ, 2016. – 188 с.

3.7. Оценка эффективности комплексной системы защиты информации

Солодянников А.В.

Прежде чем рассматривать подходы к оценке эффективности комплексной системы защиты информации, нужно определить, что такое эффективность системы. Эффективность системы – это степень достижения цели этой системой.

Целями создания КСЗИ являются предотвращение и/или минимизация воздействия угроз информационной безопасности, обеспечение непрерывности ведения бизнеса.

Эффективность КСЗИ оценивается как на этапе разработки, так и в процессе эксплуатации. В оценке эффективности КСЗИ в зависимости от используемых показателей и способов их получения можно выделить три подхода: вероятностный, оценочный, экспериментальный.

Вероятностный подход. Под вероятностным подходом к оценке эффективности понимается использование критериев эффективности, полученных с помощью показателей эффективности, полученных путем моделирования или вычисленных по характеристикам реальной системы с использованием вероятностных оценок. Такой подход используется при разработке и модернизации КСЗИ. Однако возможности вероятностных методов комплексного оценивания эффективности применительно к КСЗИ ограничены в силу ряда причин: высокая степень неопределенности исходных данных, сложность формализации процессов функционирования, отсутствие общепризнанных методик расчета показателей эффективности и выбора критериев оптимальности.

Оценочный подход. На практике часто применяется подход к оценке эффективности КСЗИ, связанный с проверкой соответствия системы защиты тем или иным требованиям. Такой подход можно условно назвать оценочным. Требования могут быть установлены государством или иным собственником информационных ресурсов.

Можно выделить две группы требований по характеру их детализации: общие и специальные. К общим относятся требования, изложенные в Федеральных законах «О персональных данных», «О коммерческой тайне», «О банках и банковской деятельности» и др. К специальным относятся требования по криптографической защите информации: технической защите информации от утечки по каналам ПЭМИН; технической защите речевой информации; защите информации от НСД в АС.

В ФЗ «О персональных данных» Правительство РФ устанавливает требования к обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных и требования

к материальным носителям биометрических персональных данных и технологиям хранения таких данных вне информационных систем персональных данных.

Контроль и надзор за выполнением требований осуществляются федеральным органом исполнительной власти, уполномоченным в области обеспечения безопасности, и федеральным органом исполнительной власти, уполномоченным в области противодействия техническим разведкам и технической защиты информации, в пределах их полномочий и без права ознакомления с персональными данными, обрабатываемыми в информационных системах персональных данных.

Согласно ФЗ «О коммерческой тайне» правообладатель имеет право установить режим коммерческой тайны, который включает:

- определение перечня информации, составляющей коммерческую тайну;
- ограничение доступа к информации, составляющей коммерческую тайну, путем установления порядка обращения с этой информацией и контроля за соблюдением такого порядка;
- учет лиц, получивших доступ к информации, составляющей коммерческую тайну, и (или) лиц, которым такая информация была предоставлена или передана;
- регулирование отношений по использованию информации, составляющей коммерческую тайну, работниками на основании трудовых договоров и контрагентами на основании гражданско- правовых договоров;
- нанесение на материальные носители (документы), содержащие информацию, составляющую коммерческую тайну, грифа «Коммерческая тайна» с указанием обладателя этой информации (для юридических лиц – полное наименование и место нахождения, для индивидуальных предпринимателей – фамилия, имя, отчество гражданина, являющегося индивидуальным предпринимателем, и место жительства).

Согласно ФЗ «О банках и банковской деятельности» за разглашение банковской тайны Банк России (организация, осуществляющая функции по обязательному страхованию вкладов), кредитные, аудиторские и иные организации, уполномоченный орган, осуществляющий меры по противодействию легализации (отмыванию) доходов, полученных преступным путем, а также их должностные лица и работники несут ответственность, включая возмещение нанесенного ущерба, в порядке, установленном федеральным законом.

Специальные требования по криптографической защите информации устанавливаются уполномоченным органом. В соответствии с Указом Президента РФ «Вопросы Федеральной службы безопасности Российской Федерации», таким органом в настоящее время является ФСБ

РФ. Основополагающим документом, в котором изложены требования по криптографической защите информации, является «Положение о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации».

Требования по технической защите информации от утечки по каналам ПЭМИН и технической защите речевой информации задаются в СТР-К и закрытых документах.

Требования по защите информации от НСД в АС могут задаваться перечнем механизмов защиты информации, которые необходимо иметь в АС, чтобы она соответствовала определенному классу защиты. Используя такие документы, можно оценить эффективность КСЗИ. В этом случае критерием эффективности КСЗИ является полнота выполнения всех требований.

Несомненным достоинством таких классификаторов (стандартов) является простота использования. Основным недостатком официального подхода к определению эффективности систем защиты является то, что эффективность конкретного механизма защиты не определяется, а констатируется лишь факт его наличия или отсутствия. Этот недостаток в какой-то мере компенсируется заданием в некоторых документах достаточно подробных требований к указанным механизмам защиты.

В Российской Федерации в настоящее время имеются две независимые системы задания требований и оценки соответствия информационных технологий требованиям безопасности информации. Наиболее распространена оценка соответствия АС требованиям Руководящего документа РД АС, СВТ – требованиям РД СВТ, межсетевых экранов – требованиям РД МСЭ. Кроме того, в рамках этого подхода для некоторых классов программного обеспечения устанавливается необходимость соответствия уровням контроля, задаваемым в РД НДВ.

Вторая система задания требований безопасности информации и оценки соответствия им основана на ГОСТ 15408–2002 и документах ФСТЭК, выпущенных в его развитие.

Требования РД СВТ и РД АС

В РД СВТ устанавливается семь классов защищенности средств вычислительной техники (СВТ) от НСД к информации. Самый низкий класс – седьмой, самый высокий – первый.

Классы подразделяются на четыре группы, отличающиеся качественным уровнем защиты:

- первая группа содержит только один седьмой класс;
- вторая группа характеризуется дискреционной защитой и содержит шестой и пятый классы;
- третья группа характеризуется мандатной защитой и содержит четвертый, третий и второй классы;

– четвертая группа характеризуется верифицированной защитой и содержит только первый класс.

Седьмой класс присваивают СВТ, к которым предъявлялись требования по защите от НСД к информации, но при оценке защищенность СВТ оказалась ниже уровня требований шестого класса.

В руководящем документе РД АС приведены требования к защищенности автоматизированных систем. В отличие от СВТ автоматизированные системы функционально ориентированы. При создании АС учитываются особенности пользовательской информации, технология обработки, хранения и передачи информации, конкретные модели нарушителя.

Устанавливается девять классов защищенности АС от НСД к информации. Классы подразделяются на три группы, отличающиеся особенностями обработки информации в АС. В третью группу входят АС, в которых имеется один пользователь, допущенный ко всей информации АС, размещенной на носителях одного уровня конфиденциальности. Группа содержит два класса – 3Б и 3А. Во вторую группу сведены многопользовательские АС, пользователи которых имеют одинаковые права доступа ко всей информации АС. Группа содержит два класса – 2Б и 2А. Первую группу составляют многопользовательские АС, в которых пользователи имеют разные права доступа к информации. Группа включает пять классов – 1Д, 1 Г, 1В. 1Б, 1А.

Для обработки конфиденциальной информации в последнем случае разрешается использовать АС классов 1Д, 1 Г.

Для примера целесообразно рассмотреть подробно требования к одному из классов защищенности, позволяющих обрабатывать конфиденциальную информацию, а именно к классу 1Д.

1. Подсистема управления доступом должна обеспечивать идентификацию и проверку подлинности субъектов доступа при входе в систему по паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов.

2. Подсистема регистрации и учета должна осуществлять регистрацию входа (выхода) субъектов доступа в систему (из системы) либо регистрацию загрузки и инициализации операционной системы и ее программного останова. Регистрация выхода из системы или останова не проводится в моменты аппаратурного отключения АС. В параметрах регистрации указываются: дата и время входа (выхода) субъекта доступа в систему (из системы) или загрузки (останова) системы; результат попытки входа: успешная или неуспешная — несанкционированная; идентификатор (код или фамилия) субъекта, предъявленный при попытке доступа. Учет всех защищаемых носителей информации должен проводиться с помощью их маркировки и занесением учетных данных в журнал

(учетную карточку), учет защищаемых носителей — в журнале (картотеке) с регистрацией их выдачи (приема).

3. Подсистема обеспечения целостности должна обеспечивать целостность программных средств СЗИ НСД, обрабатываемой информации, а также неизменность программной среды. Целостность СЗИ НСД проверяется при загрузке системы по контрольным суммам компонент СЗИ. Целостность программной среды обеспечивается использованием трансляторов с языков высокого уровня и отсутствием средств модификации объектного кода программ в процессе обработки и (или) хранения защищаемой информации. Физическая охрана СВТ (устройств и носителей информации) предусматривает контроль доступа в помещения АС посторонних лиц, наличие надежных препятствий для несанкционированного проникновения в помещения АС и хранилище носителей информации, особенно в нерабочее время. Тестирование функций СЗИ НСД при изменении программной среды и персонала АС с помощью тест-программ, имитирующих попытки НСД, проводят периодически. Следят за наличием средств восстановления СЗИ НСД, предусматривающих ведение двух копий программных средств СЗИ НСД, их периодическое обновление и контроль работоспособности.

Представленный перечень является тем минимумом требований, которым необходимо следовать, чтобы обеспечить конфиденциальность защищаемой информации.

Оценка соответствия требованиям безопасности согласно ГОСТ 15408-2002

В результате многолетней деятельности ряд развитых стран выработал «Общие критерии оценки безопасности компьютерных систем» (ОК). Документ получил статус Международного стандарта ISO/IEC в 1999 г. Гостехкомиссия приняла решение выполнить аутентичный перевод этого стандарта и принять его в качестве государственного, что и было сделано в 2002 г. С января 2004 г. данный стандарт вступил в действие.

Возможны два сценария проведения оценки: оценка уже существующего объекта оценки (ОО) и создаваемого. В первом случае считается, что ОО может быть доработан по результатам оценки на имеющийся ОО создается профиль защиты (ПЗ), содержащий общие положения по безопасности, либо ПЗ выбирается из множества существующих. Далее на основе ПЗ создается задание по безопасности (ЗБ), в котором специфицируются эти положения.

Во втором случае к ОО предъявляются требования, на соответствие которым он будет в дальнейшем проверяться. Как и в первом случае, требования предъявляются в ПЗ и ЗБ. ЗБ должно разрабатываться перед проведением оценки ОО.

Для оценки ОО очень важно определить его границы. Все, что окружает ОО, называется средой безопасности, и напрямую влияет на его без-

опасность. Выделяют программно-техническую среду, а также законодательную, административную и процедурные среды. Среда не оценивается, но относительно ее свойств делаются предположения. Отсюда следует, что, если она не удовлетворяет этим предположениям, оценка ОО теряет свое значение и тогда объект небезопасен.

В настоящее время ФСТЭК планирует создать каталог угроз, из которого разработчики могли бы выбирать актуальные для них угрозы. В тексте стандарта приведены лишь отдельные угрозы.

На основании предположений о среде формулируются цели безопасности для ОО, направленные на обеспечение противостояния угрозам и выполнение политики безопасности. В зависимости от непосредственного отношения к ОО или к среде они подразделяются на две группы. Часть целей для среды может достигаться нетехническими (процедурными) мерами. Все остальные (для объекта и среды) носят программно-технический характер. Для их достижения к объекту и среде предъявляются требования безопасности.

«Общие критерии» в главной своей части (Часть 2) как раз и являются каталогом требований безопасности. Всего перечислено 135 функциональных требований. Требования достаточно детализированы, что делает их конкретными и допускающими однозначную проверку. Большинство требований параметризовано, т.е. к ним применимы такие операции, как уточнение какого-либо значения, выбор одной возможности из нескольких.

Кроме того, к ОО могут предъявляться и нестандартные, не входящие в каталог требования, что тоже предусмотрено стандартом.

Для структуризации пространства требований в «Общих критериях...» введена иерархия: класс – семейство – компонент – элемент. Классы определяют наиболее общую (как правило, предметную) группировку требований. Семейства в пределах класса различаются по строгости и другим характеристикам требований. Компонент – минимальный набор требований, фигурирующий как целое. Элемент – неделимое требование.

Между компонентами могут существовать зависимости. Они возникают, когда компонент сам по себе недостаточен для достижения цели безопасности. Соответственно при включении такого компонента необходимо добавить всю «гроздь» его зависимостей.

Как вспомогательный элемент, упрощающий создание ПЗ и ЗБ, могут применяться функциональные пакеты (ФП) – неоднократно используемые совокупности компонентов, объединенных для достижения установленных целей безопасности.

«Общие критерии...» содержат два основных вида требований безопасности: функциональные и требования доверия. Требования доверия предъявляются к технологии и процессу разработки и эксплуатации ОО и представлены в Части 3 ОК. Сформулировав функциональные требования,

требования доверия и требования к среде, можно приступать к оценке безопасности готового изделия ИТ.

ОК предусматривают наличие нескольких уровней представления проекта с его декомпозицией и детализацией. За требованиями безопасности следует функциональная спецификация, затем проект верхнего уровня, необходимое число промежуточных уровней, проект нижнего уровня, после этого, в зависимости от типа изделия, исходный код или схемы аппаратуры и, наконец, реализация в виде исполняемых файлов, аппаратных продуктов и т. п.

При проведении оценки изделия ИТ проверяется соответствие функций безопасности ОО функциональным требованиям и корректность их реализации.

Для изделия ИТ составляется формализованный документ – задание по безопасности. В этом многостраничном документе подробно описывается не только функциональность изделия ИТ, но и его среда функционирования, угрозы, предположения безопасности, цели и требования безопасности, реализованные в изделии механизмы безопасности. В документе выполняется строгое обоснование необходимости всех реализованных механизмов. Также приводятся сведения о принятых мерах поддержки доверия.

В зависимости от принятых мер поддержки доверия (но не от набора механизмов безопасности) все изделия ИТ группируются в семь оценочных уровней доверия (ОУД). Сертификация выполняется как раз на соответствие тому или иному уровню доверия.

Сертификация изделия ИТ двухступенчатая. В начале оценивается задание по безопасности, затем само изделие – на соответствие этому заданию. За последние три года в России сертифицировано несколько изделий на соответствие ГОСТ 15408–2002.

3.8. Оценка эффективности инфраструктуры защиты информации и ее влияния на основные показатели производственно-хозяйственной деятельности предприятия

Стельмашонок Е.В., Стельмашонок В.Л.

Основным показателем экономической эффективности затрат на инфраструктуру защиты информации промышленного предприятия, как любого инвестиционного проекта является чистая приведенная стоимость (*NPV*) в период времени от *t* до *T*:

$$NPV = \sum_{t=1}^{T} \frac{\Delta if_t(R) - \Delta of_t(R)}{(1+E)^t} - K_R$$

где: $\Delta if_t(R)$ – изменение входного денежного потока с учетом проведения мероприятий по защите информации;

$\Delta of_t(R)$ – изменение выходного денежного потока с учетом проведения мероприятий по защите информации;

K_R – внеоборотные и оборотные информационные активы инфраструктуры защиты информации;

E – норма прибыли на капитал.

А теперь посмотрим, как эффективная инфраструктура защиты информации изменяет основные показатели производственно-хозяйственной деятельности предприятия.

Организация инфраструктуры защиты информации на промышленном предприятии безусловно влияет на результаты его хозяйственной деятельности.

Основными показателями хозяйственной деятельности промышленного предприятия являются:

1. основной финансовый результат (прибыль или убыток);
2. рентабельность производственных фондов и продукции;
3. стоимость предприятия (балансовый подход, доходный подход);
4. фактическая (актуальная) ликвидность.

Рассмотрим влияние затрат на инфраструктуру защиты информации на прибыль предприятия.

В результате проводимых мероприятий по защите информации на промышленном предприятии прибыль должна увеличиться по сравнению с базовым вариантом, не предусматривающим такой защиты.

$$\Delta П(R) = П(R) - П,$$

где: $\Delta П(R)$ – годовой прирост прибыли в результате мероприятий по защите информации;

$П(R)$ – прибыль при условии проведения мероприятий по защите информации за год;

$П$ – прибыль в условиях отсутствия защиты информации (базовый вариант) за год.

Затраты на инфраструктуру защиты информации содержат две составляющие: единовременные (инвестиции) и эксплуатационные (текущие).

Затраты на инфраструктуру защиты информации будут оправданы при соблюдении следующего условия:

$$\Delta П(R) \geq C_R + E \cdot K_R,$$

где: C_R – годовые эксплуатационные затраты на защиту информации.

Проводимые мероприятия по защите информации должны положительно сказаться на показателе рентабельности:

$$Q_б = \frac{П}{Ф_{пр}}$$

$$Q(R) = \frac{П + \Delta П(R) - C_R}{Ф_{пр} + K_R},$$

где: $Q_б$ – рентабельность базовая, в условиях отсутствия мероприятий по защите информации;

$Q(R)$ – рентабельность с учетом мероприятий по защите информации;

$Ф_{пр}$ – стоимость производственных фондов.

Затраты на инфраструктуру защиты информации экономически с точки зрения рентабельности оправданы при соблюдении условия:

$$\frac{П + \Delta П(R) - C_R}{Ф_{пр} + K_R} \geq \frac{П}{Ф_{пр}}.$$

Обратимся к условию эффективности затрат на инфраструктуру защиты информации на основе доходного подхода к оценке стоимости предприятия.

Как известно, приведенная стоимость предприятия (PV) учитывает временную ценность денег (в период времени от t до T):

$$PV = \sum_{t=1}^{T} \frac{if_t - of_t}{(1+E)^t} + \frac{PV_T}{(1+E)^T}.$$

С учетом создания и функционирования эффективной инфраструктуры защиты информации приведенную стоимость предприятия $PV(R)$ можно оценить:

$$PV(R) = \sum_{t=1}^{T} \frac{if_t + \Delta if_t(R) - of_t - \Delta of_t(R)}{(1+E)^t} + \frac{PV_T + \Delta PV_T(R)}{(1+E)^T},$$

где: PV – приведенная стоимость предприятия (базовый вариант) в условиях отсутствия мероприятий по защите информации;

$PV(R)$ – приведенная стоимость предприятия с учетом проведения мероприятий по защите информации;

if_t – входной денежный поток;

of_t – выходной денежный поток;

E – ставка дисконта.

Следующее условие эффективности затрат на инфраструктуру защиты информации может быть определено соотношением:

$$\sum_{t=1}^{T} \frac{if_t + \Delta if_t(R) - of_t - \Delta of_t(R)}{(1+E)^t} + \frac{PV_T + \Delta PV_T(R)}{(1+E)^T} \geq \sum_{t=1}^{T} \frac{if_t - of_t}{(1+E)^t} + \frac{PV_T}{(1+E)^T}$$

Фактическая (актуальная) ликвидность определяется величиной сальдо накопления денежных средств на расчетном счете предприятия для каждого подпериода. Очевидно, что капитальные и эксплуатационные затраты на инфраструктуру защиты информации должны не нарушать условия, при котором сальдо накопленных денежных средств на конец каждого подпериода (S_t^k) должно быть не меньше заданной величины.

$$S_t^к = (S_1^н - K_R)(1+E)^t + \sum_{\tau=1}^{t}(if_\tau + \Delta if_\tau(R))(1+E)^{t-\tau} - \sum_{\tau=1}^{t}(of_\tau + \Delta of_\tau(R))(1+E)^{t-\tau} -$$

$$- \sum_{\tau=1}^{t} of_\tau (1+E)^{t-\tau} - K \geq S_{доп\,t}^к \quad , \quad t = \overline{1,T}$$

где: $S_1^н$ – сальдо накопленных денежных средств на расчетном счете на начало первого подпериода;

if_τ – приток денежных средств в τ-ый подпериод на расчетный счет;

$\Delta if_\tau(R)$ – дополнительный приток денежных средств в τ-ый подпериод на расчетный счет в условиях функционирования системы защиты информации;

of_τ – отток денежных средств в τ-ый подпериод с расчетного счета;

$\Delta of_\tau(R)$ – дополнительный отток денежных средств в τ-ый подпериод с расчетного счета с учетом затрат на систему защиты информации;

$S_{доп\,t}^к$ – допустимый остаток денежных средств на расчетном счете предприятия на конец τ-ого подпериода.

Граничные условия эффективности затрат на инфраструктуру защиты информации обобщены в таблице 3.5.

Таблица 3.5

Граничные условия эффективности затрат на инфраструктуру защиты информации

Основные показатели хозяйственной деятельности предприятия	Граничные условия
Прибыль	$\Delta П(R) \geq C_R + E \cdot K_R$
Стоимость предприятия (доходный подход)	$\sum_{t=1}^{T} \frac{if_t + \Delta if_t(R) - of_t - \Delta of_t(R)}{(1+E)^t} + \frac{PV_T + \Delta PV_T(R)}{(1+E)^T} \geq \sum_{t=1}^{T} \frac{if_t - of_t}{(1+E)^t} + \frac{PV_T}{(1+E)^T}$
Фактическая (актуальная) ликвидность	$S_t^к = (S_1^н - K_R)(1+E)^t + \sum_{\tau=1}^{t}(if_\tau + \Delta if_\tau(R))(1+E)^{t-\tau} - \sum_{\tau=1}^{t}(of_\tau + \Delta of_\tau(R))(1+E)^{t-\tau} -$ $- \sum_{\tau=1}^{t} of_\tau (1+E)^{t-\tau} - K \geq S_{доп\,t}^к \quad , \quad t = \overline{1,T}$
Рентабельность	$\frac{П + \Delta П(R) - C_R}{Ф_{пр} + K_R} \geq \frac{П}{Ф_{пр}}$

Для определения экономической эффективности затрат на инфраструктуру защиты информации промышленного предприятия целесообразно использовать показатель чистой приведенной стоимости. При этом необходимо учитывать ценность информационных активов, которая может быть оценена методами доходного подхода, затратного подхода, а также совокупной стоимости владения (ССВ). Особенностями применения этих методов для оценки экономической эффективности инфраструктуры защиты информации являются:

- учет затрат, понесенных предприятием в результате того, что требуемый уровень информационной защищенности не будет достигнут;
- определение влияния потери информационного актива, как составного компонента предприятия, на бизнес;
- оптимизация инвестиций на защиту информации с учетом реального значения показателя совокупной стоимости владения.

ЗАКЛЮЧЕНИЕ

В монографии рассмотрены вопросы построения современных компьютерных систем с учетом требований к защите информации. Приведено теоретическое обоснование понятийного аппарата и моделей информационной безопасности. Изложены вопросы разработки систем обнаружения вторжений, анализаторов машинного кода, применения криптографических средств защиты информации, систем обнаружения вторжений.

Рассмотрены математические модели и методы управления информационной безопасностью компьютерных систем, в частности, вопросы категорирования информационных активов при формировании требований к защите информации и выбора уровня защищенности, выбора состава системы защиты информации на основе критериев экономической эффективности. Исследованы риски различных информационных систем, в том числе новых финансовых инструментов, основанных на применении информационных технологий, в частности, криптовалют и интернет-банкинга.

Приведены подходы к оценке эффективности системы защиты информации, в том числе и ее экономическая оценка в контексте производственно-хозяйственной деятельности предприятия.

Монография будет полезна специалистам по информационным технологиям и защите информации, аспирантам, магистрантам и студентам направлений «информационная безопасность», «информационные системы и технологии», «бизнес-информатика», а также других направлений при изучении дисциплин компьютерного цикла, экономическим специалистам, интересующимся применением современных информационных технологий и связанными вопросами информационной безопасности.

Научное издание

ЗАЩИТА ИНФОРМАЦИИ
В КОМПЬЮТЕРНЫХ СИСТЕМАХ

Под редакцией д-ра экон. наук Е.В. Стельмашонок,
канд. физ.-мат. наук И.Н. Васильевой

Подписано в печать 25.12.17. Формат 60×84 1/16.
Усл. печ. л. 10,25. Тираж 500 экз. Заказ 1699.

Издательство СПбГЭУ. 191023, Санкт-Петербург, Садовая ул., д. 21.

Отпечатано на полиграфической базе СПбГЭУ