и web-серверы размещены в одной защищенной частной сети (или даже на одном сетевом узле) с обслуживающими их шлюзами и только взаимодействие между шлюзами осуществляется через общедоступную сеть. Работа шлюзов основана на использовании технологии SSL/TLS [12] для надстройки защищенного канала над уже открытым TCP-соединением[13,14]. Основная идея подхода заключается в том, что, в отличие от VPN, средства защиты в данном случае подключаются на высоких уровнях иерархии протоколов OSI, что снижает универсальность, но позволяет шлюзам анализировать высокоуровневые параметры информационных запросов и ответов web-серверов, содержащихся в http-заголовках. А это, в свою очередь позволяет добавить в шлюзы дополнительный «интеллект», связанный с проверкой подлинности серверов и клиентов, с маршрутизацией запросов в мульти-серверной среде, а также с разграничением доступа к информационным ресурсам с точностью до отдельных функций (методов) в составе web-сервисов на основе реквизитов владельцев, содержащихся в сертификатах открытого ключа. Важно подчеркнуть, что эти дополнительные возможности не «привязаны» к потребностям конкретного проекта, но реализованы в форме готовых к использованию «общих» решений. Известные подходы к построению защищенных сетевых каналов взаимодействия не предоставляют таких возможностей.

1 Основы организации защищенного канала

Работа защищенного организована в соответствии со следующими основными положениями.

- Защищенный канал строится на основе двух базисных технологий: технологии SSL/TLS и технологии proxy-серверов. Оба шлюза - и клиентский и серверный – по своей роли и статусу в системе представляют собой proxy-серверы – постоянно активные программы («демоны») выполняющие функции посредников между клиентскими и сервисными компонентами РС. В данном случае цель «посредничества» заключается в защите данных в сети.
- Объектами и «единицами» обработки в канале являются не отдельные IP-пакеты, но цельные электронные документы: информационные запросы к web-сервисам и результаты их обработки (ответы) в формате HTTP/SOAP.
- Главную роль в организации защиты данных в защищенном канале играют сертификаты открытого ключа клиентского и серверного шлюзов. Хотя в современных версиях протоколов SSL/TLS передача сертификата клиента серверу, вообще говоря, является опциональной, в данном случае она является обязательной: если в результате процедуры «рукопожатия» (handshaking) сертификат клиентского шлюза не был получен, серверный шлюз немедленно разрывает соединение с последним.
- Средства защиты опираются на функциональные возможности библиотек libssl и libcrypt для ОС Linux и реализованы в рамках защищенного канала. Ни клиентские, ни серверные компоненты РС не обязаны иметь дело ни с крипто-функциями, ни с сертификатами или закрытыми ключами. В частности, средства авторизации, основанные на сертификатах, должны быть поддержаны в самом канале, а не в компонентах РС.
- Каждый клиентский шлюз способен взаимодействовать не с одним, но со многими серверными шлюзами. Выбор серверного шлюза осуществляется на основе Интернет-имени адресуемого web-сервера с использованием специальной таблицы маршрутизации, содержащейся в конфигурационном файле клиентского шлюза и связывающей серверные шлюзы с адресуемыми web-сервисами. Важная особенность описываемого подхода заключается в том, что функции маршрутизации «переплетаются» здесь с функциями защиты: упомянутая таблица содержит не только Интернет-имена (или адреса) серверных шлюзов, но и требования к их сертификатам, сформулированные в терминах ограничений на реквизиты владельца (защита от сфальцифицированного серверного шлюза).

С точки зрения администратора РС самой важной компонентой шлюза является его конфигурационный файл ssltunnel.cfg, который содержит параметры (настройки), управляющие его работой. Параметры задаются в форме: *Ключевое слово = значение*

Каждая настройка размещается в отдельной строке. Основные параметры и связанные с ними ключевые слова перечислены в таблице 1 .

Первые четыре параметра являются вполне характерными для сетевых серверных программ. Сертификат и закрытый ключ шлюза загружаются из файлов с расширением ".pem" с помощью функций SSL_CTX_use_certificate_file() и SSL_CTX_use_Privatekey_file() библиотеки libssl соответственно (они могут быть подготовлены с помощью утилиты openssl). Отметим, что вместо имен файлов могут быть указаны имена директорий: в этом случае шлюз осуществит поиск файлов в

соответствующей директории и ее поддиректориях. Сертификат открытого ключа будет загружается из первого же найденного файла, имя которого заканчивается на «_cert.pem» (например, My_cert.pem), а закрытый ключ - из первого же найденного файла, имя которого заканчивается на «_key.pem» (например, My_key.pem). Такая организация позволяет считывать сертификаты и ключи со съемных носителей, задавая имя директории, в которую они монтируются (например, "/media").

Таблица 1. Основные настройки шлюза защищенного канала

№	Ключевое слово	Настройка	Значение по умолчанию
1	ListenPort	Номер порта для входящих соединений	8128
2	ListenIp	IP-адрес (или имя) сетевой карты	127.0.0.1
3	ReadTimeout	Таймаут чтения из сокета (сек.)	180
4	WriteTimeout	Таймаут записи в сокет (сек.)	60
5	CertificateFile	Полное имя файла сертификата открытого ключа шлюза (или имя директории для поиска)	/media
6	PrivateKeyFile	Полное имя файла закрытого ключа шлюза (или имя директории для поиска)	/media
7	TrustedDir	Полное имя директории доверенных сертификатов	etc/ssl/certs

Кроме параметров, указанных в таблице 1, конфигурационный файл может содержать разделы, специфичные для клиентского и серверного шлюзов. Эти разделы прямо относятся к поиску и проверке подлинности серверного шлюза (раздел [Forward] для клиентского шлюза) или к проверке прав клиента на доступ к web-сервисам и отдельным сервисным функциям (раздел [Access] для серверного шлюза). Лучше всего пояснить структуру этих разделов на конкретном примере.

Рассмотрим систему управления большой организацией Titan, имеющей центральный офис в Москве и филиалы в областных центрах страны. Предположим, что на web-серверах центрального офиса и офиса каждого филиала имеется web-сервис "Staff", который включает несколько сервисных функций для регистрации и учета служащих этого филиала. К их числу относится функции AddPerson, DeletePerson и CorrectPerson, обеспечивающих добавление, удаление и коррекцию записей о служащих в БД филиала, и функция ViewPersons, позволяющая получить список служащих офиса. Предположим, также, что на web-серверах каждого офиса имеется web-сервис "Stat", который содержит функцию Report, позволяющую сформировать статистический отчет о кадрах филиала за любой период времени (например, о динамике роста числа служащих в разрезе подразделений). Будем считать, что рабочие места служащих компании оснащены рабочими станциями, позволяющими обращаться к web-сервисам компании дистанционно через Интернет с использованием рассматриваемых защищенных HTTPS-каналов. Будем, также, считать, что в целях безопасности web-серверы каждого офиса размещены в его частной сети. Другими словами, доступ к этим сервисам из Интернета осуществляется через серверный шлюз защищенного канала, размещенный на выделенном сервере, подключенных и к частной сети офиса, и к Интернету через два сетевых интерфейса. Предположим, что кампания снабжает каждого служащего личным закрытым ключом и сертификатом открытого ключа, удостоверенным электронной подписью центрального офиса. Файл сертификата центрального офиса имеется на всех рабочих станциях и серверных шлюзах в директории «доверенных» сертификатов /etc/ssl/certs/main_office.

На рис. 1 приведен пример конфигурационного файла клиентского шлюза (размещенного на клиентской рабочей станции), а на рис. 2 – пример конфигурационного файла серверного шлюза.

```
ListenPort=8128
CertificateFile=/media
PrivateKeyFile=/media
TrustedDir=/etc/ssl/certs/main_office

[Forward]
www.titan.moscow.ru 193.182.12.1  C=Russia,L=Moscow,O=Titan,CN=TitanSrv
www.titan.spb.ru 47.172.33.2  C=Russia,L=StPeterburg,O=Titan,CN=TitanSrv
. . .
www.titan.omsk.ru 146.58.130.3:8443  C=Russia,L=Omsk,O=Titan,CN=TitanSrv
```

Рис. 1. Пример конфигурационного файла клиентского шлюза

Как видно из рис. 1, в разделе [Forward] клиентского шлюза размещена последовательность строк, каждая из которых содержит описание одного серверного шлюза. Это описание связывает Интернет-имя адресуемого web-сервера с основными характеристиками обслуживающего его серверного шлюза: IP-адресом (или Интернет-именем) и требованиями к реквизитам владельца в формате, соответствующем стандарту X509. Эти требования используются для проверки подлинности серверного шлюза, которая проводится сразу же после получения его сертификата открытого ключа. Предположим, например, что клиентская программа выполняет вызов web-сервиса с URL, равным http://www.titan.spb.ru/staff.asmx. Тогда, в соответствии со второй строкой в разделе [Forward]:

- клиентский шлюз защищенного канала выполнит сетевое соединение с серверным шлюзом с IP-адресом 47.172.33.2 (порт 443 предполагается по умолчанию) и инициирует создание защищенного канала в соответствии с технологией SSL/TLS (см. рис. 3),
- если поле Sabject Name в сертификате, полученном от серверного шлюза в результате процедуры «рукопожатия» (handshaking), будет содержать реквизиты владельца, отличные от C=Russia,L=StPeterburg, O=Titan,CN=TitanSrv (страна Russia, город StPeterburg, организация Titan, имя/название TitanSrv), то клиентская программа получит сообщение о сфальсифицированном серверном шлюзе, и соединение с последнем будет немедленно разорвано.

Важно подчеркнуть, что клиентский шлюз использует раздел [Forward] как своего рода «таблицу маршрутизации» информационных запросов в сети; т.е. для определения адреса серверного шлюза, которому должен быть направлен запрос через защищенное сетевое соединение (рис. 3).

Как видно из рис. 2, раздел [Access] в серверном шлюзе содержит описания ограничений доступа к web-сервисам санкт-петербургского филиала с именами www.titan.spb.ru/stuff.asmx и www.titan.spb.ru/stat.asmx. Кроме имени сервиса каждое описание может содержать требования к реквизитам клиентов, которым разрешено обращаться к данному сервису в уже знакомой нам форме.

```
ListenPort=443
ListenIp=47.172.33.2
CertificateFile=/media
PrivateKeyFile=/media
TrustedDir=/etc/ssl/certs/main_office

[Access]
www.titan.spb.ru/staff.asmx C=Russia,L=StPeterburg,O=Titan
    AddPerson   C=Russia,L=StPeterburg,O=Titan,OU=Inform Department
    DeletePerson C=Russia,L=StPeterburg,O=Titan,OU=Inform Department
    CorrPerson C=Russia,L=StPeterburg,O=Titan,OU=Inform Department
    ViewPersons
www.titan.spb.ru/stat.asmx
    Report  C=Russia,L=StPeterburg,O=Titan,CN=Admin*:* |
        C=Russia,O=Titan,CN=Top*:*
```

Рис. 2. Пример конфигурационного файла серверного шлюза

На этих требованиях основана процедура проверки прав доступа к web-сервисам: содержание поля Subject Name в сертификате, полученном от клиентского шлюза в результате процедуры «рукопожатия» (handshaking), должно им соответствовать. После, строки, содержащей имя web-сервиса, могут следовать строки, содержащие имена отдельных функций. Эти строки также могут содержать требования к реквизитам клиентов, специфичные для той или иной функции. Эти требования «приоритетнее» требований, содержащихся в описании web-сервиса. В приведенном примере только служащие санкт-петербургского офиса из подразделения «Inform Department» имеют право вызывать функции AddPerson, DeletePerson и CorrPerson, а функция ViewPerson доступна любому служащему офиса.

В нижней части рис. 2 приведено описание ограничений доступа к функции Report. Они определяются несколько сложнее. В данном примере предполагается, что в соответствии с корпоративным стандартом реквизит CN (Common Name) в сертификате служащего должен быть задан в формате «должность: имя». Как видно из рис. 2, право доступа к функции имеют служащие двух категорий: работники санкт-петербургского офиса, название должности которых начинается со слова “Admn” (символ «*» означает «любая подстрока»), а также работники любого офиса компании, название должности которых начинается со слова “Top” (символ «|» в данном случае означает логическую связку «или»).

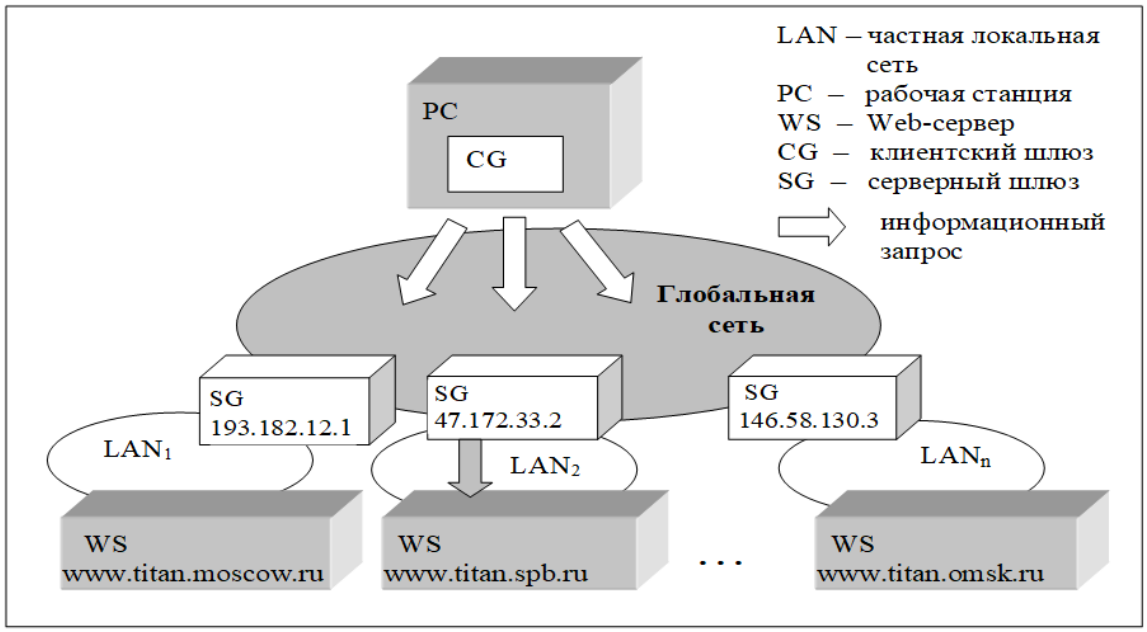

Рис. 3. Маршрутизация защищенных информационных запросов

Диагностические сообщения о сфальсифицированном сервере или о нарушении прав доступа к сервисам передаются в программу клиента в форме HTTP-ответа с кодом 500 (Internal server error), содержащего текст сообщения в формате SOAP. Этот ответ создает исключительную ситуацию в программе клиента как результат попытки обращения к web-сервису. Если вызов сервисной функции заключен в блок «try – catch», то программа клиента легко может получить текст сообщения из параметра предложения catch. Разумеется, все это справедливо и в отношении любых других диагностических сообщений от шлюзов. Эти сообщения могут быть связаны с неудачной попыткой сетевого соединения клиентского шлюза с серверным, с ошибкой чтения из сокета или записи в сокет, с некорректным сертификатом и т.п. Диагностические сообщения, порожденные клиентским шлюзом, передаются непосредственно программе клиента. Диагностические сообщения, порожденные серверным шлюзом, передаются сначала в клиентский шлюз, а оттуда программе клиента.

2 Временные оценки

Как видно из вышеприведенного описания, логика работы защищенного канала предполагает включение двух промежуточных серверов-шлюзов между клиентской программой и web-сервером. Такая организация вызывает естественный вопрос о величине неизбежной дополнительной задержки в обработке запросов к web-серверу. С целью оценки этой задержки была выполнена

экспериментальная реализация защищенного канала на основе библиотек libssl и libcrypt (пакет libssl-dev версии 1.1.1d) в Linux Debian 10 на языке C++ и проведена серия экспериментов в лабораторной сети с использованием web-сервера apache2 версии 2.4.38.

Поскольку величина дополнительной задержки при вызове web-сервиса, очевидно, зависит от размеров информационного запроса и результата его выполнения (сетевого трафика), были использованы несколько сервисных функций с различным объемом передаваемых данных в обоих направлениях и различными временами обработки. На рис. 4 приведены времена обращения к сервисной функции с малым временем обработки (100 мс) для различных объемов передаваемых данных (10 Кб и 200 Кб) в двух режимах: без защиты («напрямую») и через защищенный канал. На рис. 5 приведены времена обращения в тех же условиях к более «медленной» функции (время обработки 500 мс).

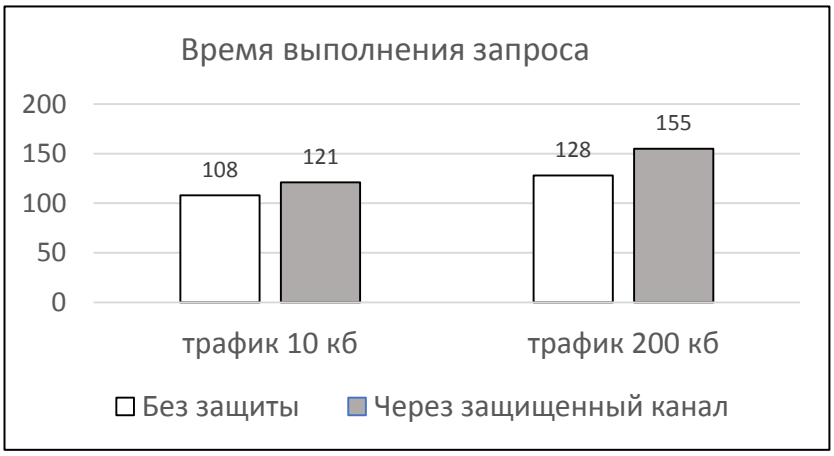

Рис. 4. Времена выполнения запроса к «быстрой» функции (100 мс)

Как видно из рис. 4 и 5, в результате подключения защищенного канала время выполнения запроса к функции с временем обработки 100 мс увеличилось на 13 мс (12%) при трафике 10 Кб, а при трафике 200 Кб – на 27 мс (21%). При вызове же функции с временем обработки 500 мс дополнительная задержка составила 14 мс (2.7%) при трафике 10 Кб и 29 мс (5.5%) при трафике 200 Кб. Как и следовало ожидать, если в первом случае потери быстродействия могут быть достаточно велики, то во втором случае они выглядят значительно более умеренными.

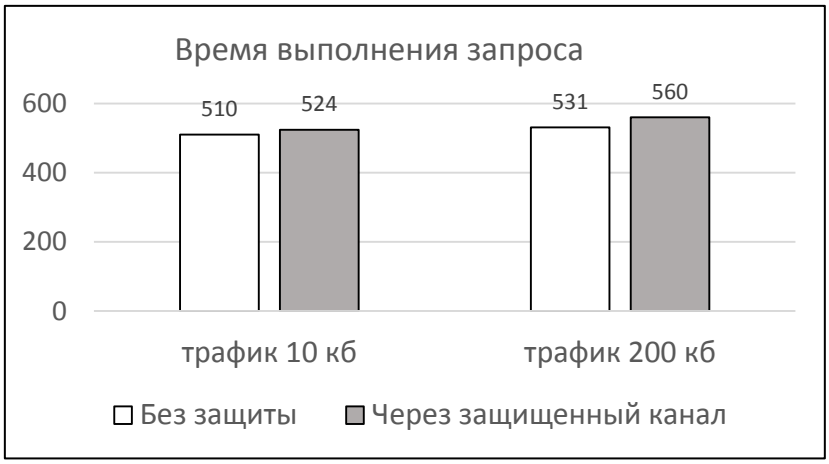

Рис. 5. Времена выполнения запроса к более «медленной» функции (500 мс)

Если на рис. 4 и 5 отображены результаты экспериментов в режиме одиночных запросов, то на рис. 6 приведены результаты оценки быстродействия защищенного канала в условиях высокой нагрузки: при параллельной обработке пакетов одновременно поступивших информационных запросов. Кривые, представленные на рисунке, отражают характерную зависимость среднего времени выполнения запроса от их количества в пакете при вызове «быстрой» сервисной функции со временем обработки 100 мс и объемом передаваемых данных 100 Кб в двух режимах: без защиты («напрямую») и через защищенный канал. Как видно из графика, обе кривые демонстрируют устойчивый рост, который объясняется ограниченной производительностью шлюзов и web-сервиса и

повышением накладных расходов на управление параллельными обрабатывающими «нитями» (программными потоками) по мере увеличения числа запросов в пакете. Однако, относительное увеличение среднего времени выполнения запроса вследствие подключения защищенного канала не превышает 15%.

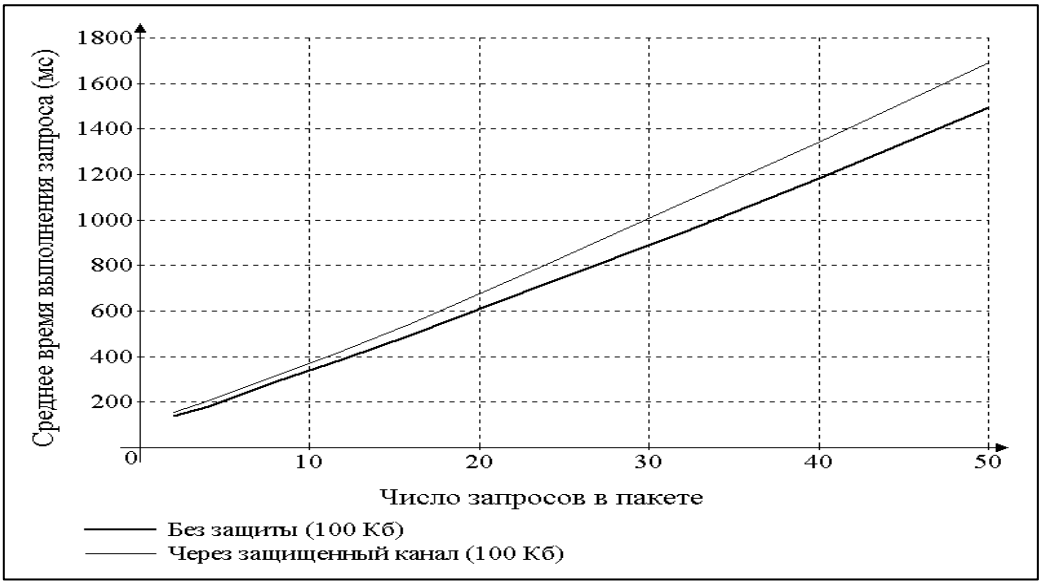

Рис. 6. Среднее временя выполнения запроса в пакете одновременных запросов

Как видно из приведенных результатов, при вызове «быстрых» сервисных функций с временем обработки 0.1 секунды относительные потери быстродействия могут быть существенными, особенно при большом объеме передаваемых данных (рис. 4). Однако, при увеличении времени обработки сервисной функции до 0.5 секунды относительные потери снижаются до нескольких процентов даже при трафике 200 Кб (рис. 5). По-видимому, именно сервисные функции со временем обработки 0.5 секунды и выше составляют область наиболее эффективного применения описанного подхода.

Заключение

Разумеется, при разработке любой РС как правило закладываются и реализуются те или иные средства аутентификации и авторизации еще на уровне клиентских компонент. Описанный подход к организации проверки и разграничения прав доступа к сервисам и сервисным функциям в рамках защищенного сетевого канала позволяет наложить общие ограничения на поведение любых клиентов, использующих этот канал. Это особенно важно в тех случаях, когда потребителям информации предоставляется возможность разработки собственных клиентских компонент для доступа к web-сервисам и информационным ресурсам.

Описанный защищенный канал в первую очередь ориентирован на РС, разрабатываемые на языке C# в среде программирования MonoDevelop. Именно в одном из таких проектов он и прошел «боевое крещение». Важно отметить, что этот канал может быть использован не только в клиентских компонентах РС для вызова функций web-сервисов, но и в служебных программах. Например, утилита wsdl, широко используемая для считывания формализованного описания web-сервиса и автоматической генерации исходного модуля прокси-класса, вполне уверенно работает через описанный канал (при условии, что в системных сетевых настройках указаны адрес и порт клиентского шлюза в качестве параметров proxy-сервера). Другими словами, предложенная технология может быть использована не только при эксплуатации РС, но и при их разработке.

В имеющейся на сегодняшний день реализации защищенного канала функции клиентского и сервисного шлюзов выполняются одной и той же программой (разница только в конфигурационных настройках). Шлюзы приспособлены к работе в режиме «демонов» - невидимых фоновых программ, разрывающих связи с монитором, клавиатурой и мышью сразу после запуска. Во время работы шлюз занимает всего около 1 Мб оперативной памяти.

Литература

1. *Козлов А.Д., Орлов В.Л.* Методы и средства обеспечения информационной безопасности распределенных корпоративных систем. – М.: ИПУ РАН, 2017. – 156 с.

2. *Салимова Ш.А.* Кибербезопасность в России: актуальные угрозы и пути обеспечения в современных условиях // Достижения вузовской науки 2021: сборник статей XVII Международного научно-исследовательского конкурса, Пенза, 20 января 2021 года. – Пенза: "Наука и Просвещение", 2021.– С. 207 - 214.
3. *Жаранова А.О., Птицына Л.К.* Анализ влияния распределенности на качество функционирования комплексных систем защиты информации // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020): Сборник научных статей IX Международной научно-технической и научно-методической конференции. – СПб: СПБГУТ, 2020. – С. 324-327.
4. *Згоба А.И., Маркелов Д.В., Смирнов П.И.* Кибербезопасность: угрозы, вызовы, решения / Вопросы кибербезопасности, № 5, 2014. – С. 30 – 38.
5 *Шапошников И.В.* Web-сервисы Microsoft .NET. – СПб: БХВ-Петербург, 2002. – 336 с.
6. *Мак-Дональд М., Шпушта М.* Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. – М.: Вильямс, 2009. – 1408 с.
7 *Liu M.* WCF multi-layer services development with entity framework. – Birmingham: Packt Publishing, 2014. – 378 p.
8. *Lowy J., Montgomery M.* Programming WCF services: design and build maintainable service-oriented systems. – N.Y.:O'Reilly Media, 2015. – 1018 p.
9. *Костин Е. И.* Создание службы WCF для использования в клиент-серверном приложении // Инновационные подходы в решении научных проблем : Сборник трудов по материалам IV Международного конкурса научно-исследовательских работ, Уфа, 01 марта 2021 года. – Уфа: Общество с ограниченной ответственностью "Научно-издательский центр "Вестник науки", 2021. – С. 121-126.
10. *Negus C.* Linux Bible N.J., Wiley, 2015, 912 p.
11. *Акушуев Р. Т.* Принцип работы VPN и его особенности // Modern Science, № 7, 2020. – С. 312-314.
12. *Baka P., Schatten J.* SSL/TLS under lock and key: a guide to understanding SSL/TLS cryptography. – Keyko books, 2020. – 132 p.
13. *Снейдер Й.* Эффективное программирование TCP/IP. Библиотека программиста. – СПб.: Символ-Плюс, 2002. – 20 с.
14. *Хант К.* TCP/IP. Сетевое администрирование. – СПб.: Питер, 2007. – 816 с.