

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  **информационных систем**  **и технологий** | **Кафедра**  **информационных систем** |

**Уланович Илья Дмитриевич**

**Тема**: **«Визуализция серверной инфраструктуры телекоммуникационной компании на основе докер-контейнеров »**

**Выпускная квалификационная работа на присвоение квалификации «бакалавр» по направлению 09.03.02 «Информационные системы и технологии»**

|  |  |  |
| --- | --- | --- |
| Заведующий кафедрой  д.т.н., проф. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Позднеев Б.М.**  подпись |  |
| Научный руководитель  к.т.н., доц. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Тясто С.А.**  подпись |  |
| Студент  группы ИДБ-15-13 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Уланович.И.**  подпись |  |
|  |  |  |
|  |  |  |

Регистрационный № \_\_\_\_\_\_\_\_

[ВВЕДЕНИЕ 2](#_Toc34492254)

ВВЕДЕНИЕ

Стремительное развитие информационных технологий вывело организационную структуру предприятий на новый уровень. На сегодняшний день каждая даже самая малая компания обязана иметь свои сервера и какую то либо бы базу данных и это не говоря о том, что многие компании не могут существовать без такой структуры.

Серверная инфраструктура — специализированное оборудование, предназначенное для сбора, обработки и хранения большого количества информации. На устройства устанавливается специализированный софт, который обеспечит функционирование предприятия или компании. Сбои в работе данных систем парализуют работу фирмы или предприятия. Так же в зависимости от важности хранения данных необходимо обеспечить должный уровень защиты. А если на этих серверах должны храниться персональные данные пользователей, то нужно не просто обеспечить информационную безопасность, а получить разрешения у фсб на хранение в данном месте данные такого типа.

Нормальной практикой для компаний считается иметь свой штат системных администраторов, которые вводят в строй новые сервера, устанавливают обновления, решают проблемы с серверами и т.д. Чем шире перечень серверных операционных систем, тем больше системных администраторов необходимо иметь в штате.

Для решения задач поддержки серверной инфраструктуры, как правило, необходимо обеспечить круглосуточный мониторинг и обслуживание серверов, а также организовать группу специалистов для восстановления работоспособности оборудования. Обычно эти мероприятия требуют выделения большого объема ресурсов, что дорого и не всегда эффективно.

При рассмотрении вопросов поддержки серверной инфраструктуры ключевым становится разработка и реализация серверной архитектуре компании - чем проще архитектура (однотипные устройства, стандартизированные системы, описанные настройки и т.д.), тем меньше нужно ресурсов на ее поддержку, а значит – тем меньше затраты на поддержку.

Особое место в модернизации существующей серверной архитектуры и поддержке серверной инфраструктуры занимает виртуализация. В данной дипломной работе будут использоваться сторонние открытые ресурсы виртуализации draw.io в котором будет о отрисована вся структура на прикладном уровне и Portainer. Portainer — это удобное управленческое решение для докеров, позволяющее визуализировать и управлять ими с помощью удобного веб интерфейса.

Таким образом, была поставлена цель автоматизации серверной инфраструктуры телекоммуникационной компании на основе докер-контейнеров

Для достижения поставленной цели, необходимо решить следующие задачи:

1. Проанализировать современное open source программное обеспечение для маршрутизации. Выбрать из имеющихся сборок одну на dockehub;
2. Визуализировать структуру которую необходимо будет построить;
3. Запустить вручную один контейнер;
4. Автоматизировать контроль работы контейнеров в yml файле;
5. Произвести тестовое подключение к внешнему оператору;

Глава1. Анализ современных средств программной маршрутизации на основе докер-контейнеров.

1.1. Основные характеристики средств программной маршрутизации.

Программная маршрутизация выполняется либо специализированным ПО маршрутизаторов (в случае, когда аппаратные методы не могут быть использованы, например, в случае организации туннелей), либо программным обеспечением на компьютере. В общем случае, любой компьютер осуществляет маршрутизацию своих собственных исходящих пакетов (как минимум, для разделения пакетов, отправляемых на шлюз по умолчанию и пакетов, предназначенных узлам в локальном сегменте сети). Для маршрутизации чужих IP-пакетов, а также построения таблиц маршрутизации используется различное ПО:

Сервис RRAS ([анг.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) routing and remote access service) в Windows Server

Домены routed, gated, quagga в Unix-подобных операционных системах (Linux, FreeBSD и т. д.)

Маршрутизация в сети Интернет основана на протоколах TCP/IP.

Передача информации осуществляется с помощью IP-пакетов, заголовок каждого IP-пакета содержит IP-адрес получателя и отправителя пакета. Каждый пакет обрабатывается маршрутизатором в соответствии с его таблицей маршрутизации. Таблица, в свою очередь, содержит информацию, компьютеру с каким адресом направлять пакеты с тем или иным диапазоном адресов. Например, все пакеты определённого диапазона могут направляться другому маршрутизатору, который «отвечает» за этот сегмент.

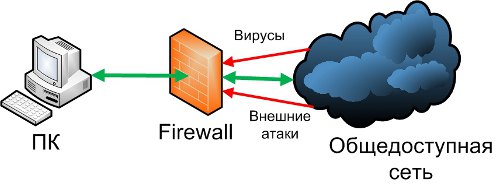
В ряде случаев маршрутизатор может преобразовывать заголовок пакета, заменяя адреса отправителя и/или получателя пакета. В частности, это происходит при взаимодействии локальной сети (имеющей свои адреса) с глобальной сетью Интернет. В этом случае локальная сеть может быть видна извне по одному глобальному IP-адресу. Для того, чтобы маршрутизатор мог направлять пакеты с одним глобальным адресом тем или иным получателям в локальной сети, используется таблица NAT, где помимо IP-адресов указываются порты, идентифицирующие приложения, устанавливающие соединение. При этом номера портов указаны не в заголовке IP-пакета, а в заголовке сегмента TCP либо UDP (сегменты инкапсулируются в поле данных IP-пакетов). Это позволяет осуществлять взаимно-однозначную идентификацию получателя и отправителя в тех случаях, когда за одним глобальным адресом находится множество компьютеров локальных сетей.

Пример таблицы NAT:

|  |  |
| --- | --- |
| 209.165.200.226:1444 | 192.168.1.15:1444 |
| 209.165.200.226:1445 | 192.168.1.16:1444 |

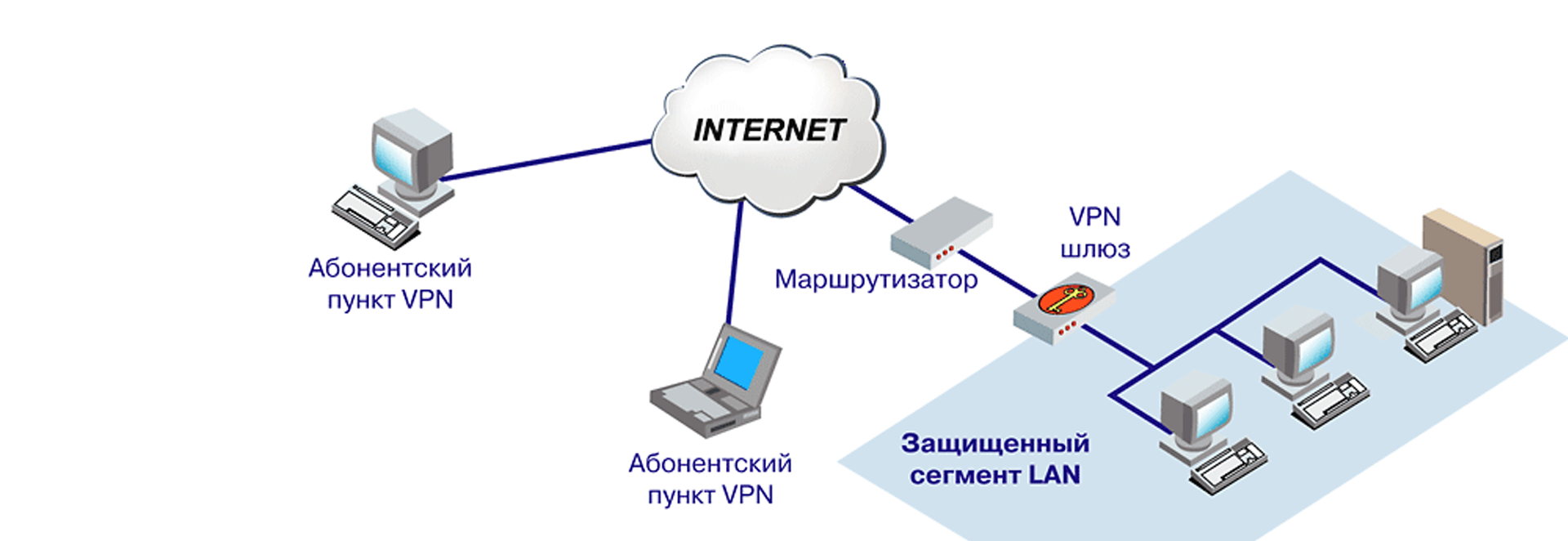
Межсетево́й экра́н, сетево́й экра́н — программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами.

Среди задач, которые решают межсетевые экраны, основной является защита сегментов сети или отдельных хостов от несанкционированного доступа с использованием уязвимых мест в протоколах сетевой модели OSI или в программном обеспечении, установленном на компьютерах сети. Межсетевые экраны пропускают или запрещают трафик, сравнивая его характеристики с заданными шаблонами.



Наиболее распространённое место для установки межсетевых экранов — граница периметра локальной сети для защиты внутренних хостов от атак извне. Однако атаки могут начинаться и с внутренних узлов — в этом случае, если атакуемый хост расположен в той же сети, трафик не пересечёт границу сетевого периметра, и межсетевой экран не будет задействован. Поэтому в настоящее время межсетевые экраны размещают не только на границе, но и между различными сегментами сети, что обеспечивает дополнительный уровень безопасности.

VPN ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Virtual Private Network «виртуальная частная сеть») — обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)). Несмотря на то, что коммуникации осуществляются по сетям с меньшим или неизвестным уровнем доверия (например по публичным сетям) уровень доверия к построенной логической сети не зависит от уровня доверия к базовым сетям благодаря использованию средств криптографии (шифрования, аутентификации, [инфраструктуры открытых ключей](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D1%80%D0%B0%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D1%85_%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%B9), средств для защиты от повторов и изменений передаваемых по логической сети сообщений).



При должном уровне реализации и использовании специального программного обеспечения сеть VPN может обеспечить высокий уровень шифрования передаваемой информации. При правильной настройке всех компонентов технология VPN обеспечивает [анонимность](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%BE%D1%81%D1%82%D1%8C) в Сети.

В данном проекте ПО маршрутизатор должен обладать следующим функционалом.

* межсетевой экран для IPv4 и IPv6, включая фильтрацию p2p-трафика;  
  трансляция сетевых адресов (NAT);
* система обнаружения вторжений;
* балансировка нагрузки и резервирование канала;
* резервирование маршрутизаторов с синхронизацией таблицы состояний соединений;
* виртуальные частные сети (IPsec, L2TP/IPsec, PPTP, OpenVPN);
* учёт трафика (Netflow и sFlow);
* Система единой конфигурации.
* Бесплатен.

1.2. Существующие системы.

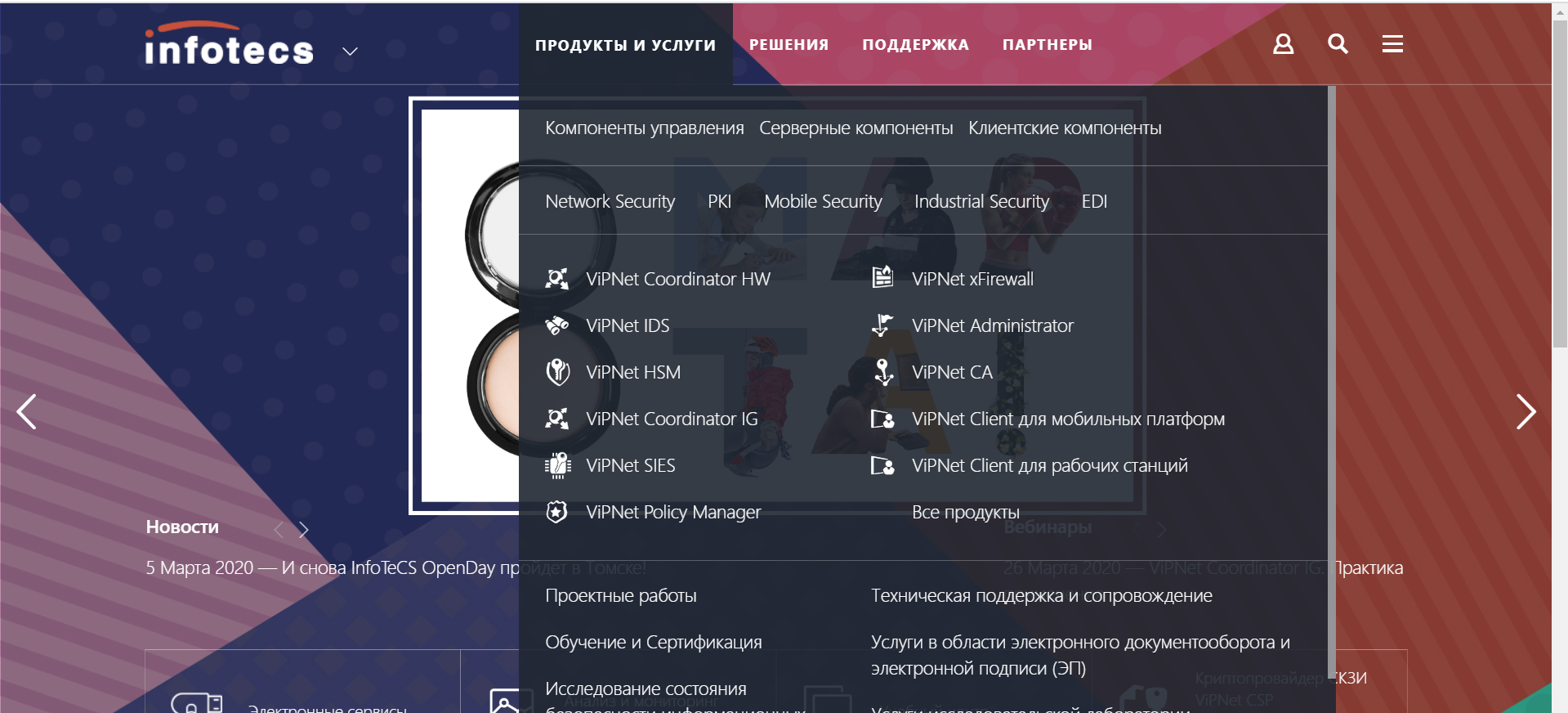
На данный момент существует множество основных решений программных межсетевых экранов: ViPNet Coordinator, Strongswan, Vyos.

Начнем с ViPNet CoordinatoR- семейство шлюзов безопасности, входящих в состав продуктовой линейки ViPNet Network Security.

Это сертифицированный продукт позволяет производить все необходимые нам настройки и поддерживает высокую безопасность сети при должном контроле. Благодаря тому что он имеет такие сертификаты как:

* [Сертификат соответствия ФСБ России № СФ/124-3657 от 20.03.2019](https://infotecs.ru/include/pages/withpdf.detail.php?ELEMENT_ID=12082&IBLOCK_ID=14)
* [Сертификат соответствия ФСБ России №СФ/118-3510](https://infotecs.ru/include/pages/withpdf.detail.php?ELEMENT_ID=11876&IBLOCK_ID=14)
* [Сертификат соответствия ФСБ России № СФ/124-3430 ПК ViPNet Client 4](https://infotecs.ru/include/pages/withpdf.detail.php?ELEMENT_ID=11771&IBLOCK_ID=14)
* [Сертификат соответствия ФСБ России № СФ/124-3431 на ПК ViPNet Coordinator 4](https://infotecs.ru/include/pages/withpdf.detail.php?ELEMENT_ID=11907&IBLOCK_ID=14)
* [Сертификат соответствия ФСБ России № СФ/124-3563 на программный комплекс ViPNet Coordinator 4 for Linux](https://infotecs.ru/include/pages/withpdf.detail.php?ELEMENT_ID=11936&IBLOCK_ID=14)
* [Сертификат соответствия ФСБ России № СФ/124-3674 на ПАК ViPNet Coordinator HW 4](https://infotecs.ru/include/pages/withpdf.detail.php?ELEMENT_ID=12140&IBLOCK_ID=14)

Компания INFOTECS может просить за этот продукт большие деньги для подключение к государственным структурам. Поэтому по факторы цены данный кандидат не подходит.



Следующий кандидат это Strongswan. StrongSwan является демоном IPSEC, который поддерживает IKEv1 и IKEv2. На данный момент это развивающий продукт. Установка StrongSwan может быть выполнена из исходников или репазитория.



Файлы конфигурирования по умолчанию хранятся в директории /etc/ и имеют следующие названия:

ipsec.conf – определяет параметры IPSEC-соединений и параметры подключений в целом;

ipsec.secrets – служит для хранения ссылок на сертификаты и ключи аутентификации;

strongswan.conf – для подключения криптографических алгоритмов и дополнительных функций.  
Помимо этого во время установки программного обеспечения для хранения сертификатов и CRL-файлов используемых демонами pluto и charon создается директория /etc/ipsec.d, в которой находятся следующие каталоги:

* private – содержит закрытые ключи RSA и ECDSA;
* certs – содержит сертификаты X.509 и PGP;
* crls – хранит список отозванных сертиифкатов;
* cacerts – хранит доверенные сертификаты CA;
* ocspcerts – содержит подписанные OCSP сертификаты;
* reqs – содержит запросы на сертификаты в формате PKCS#10.

Файл /etc/ipsec.secrets содержит неограниченное количество следующих типов ключей (паролей):RSA для определения пароля к сертификату открытого ключа;ECDS для определения пароля к сертификату открытого ключа;

PSK для определения Pre-shared ключа;

EAP для учетных записей EAP;

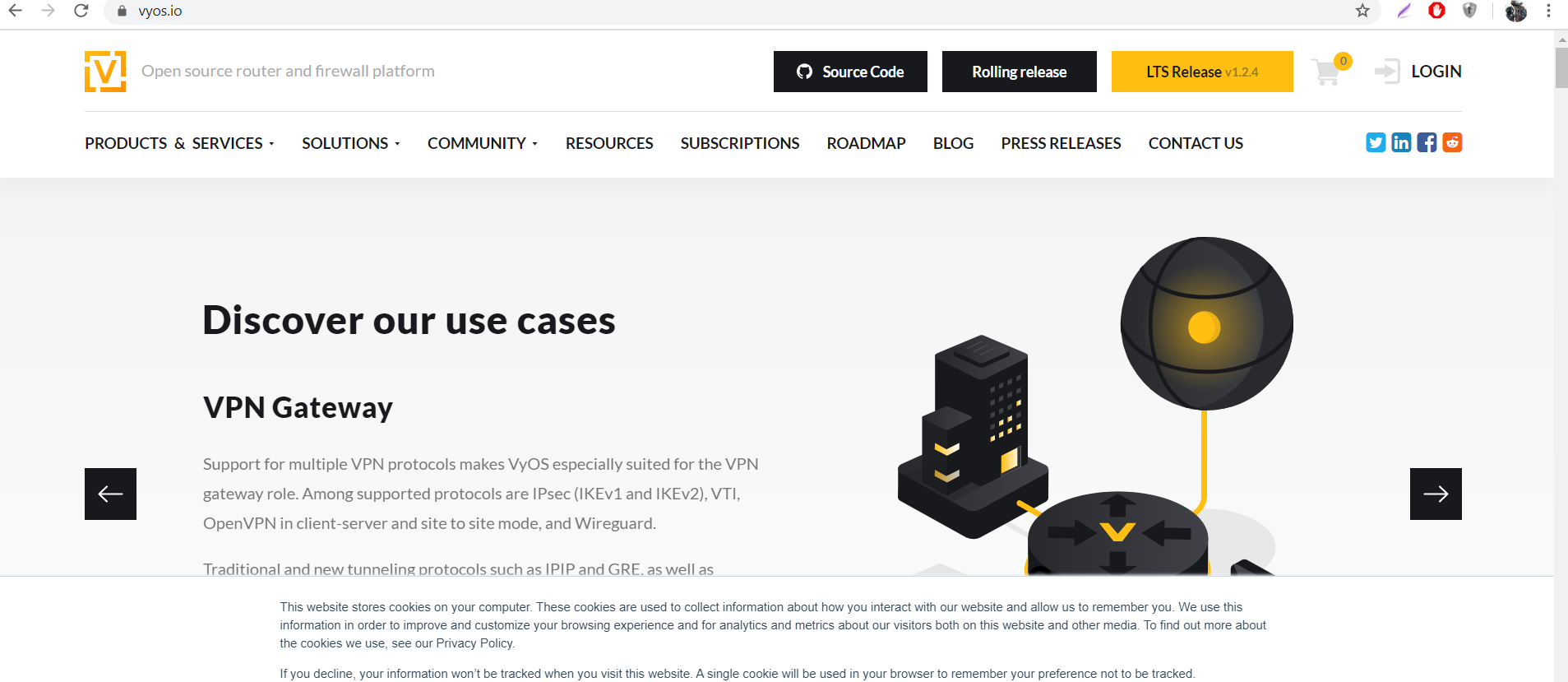
NTLM для учетных записей NTLM;

XAUTH для учетных записей XAUTH;

PIN для пин-кода смарт-карт.

Соответственно поддерживаются все типы аутентификации.  
Основные параметры команды ipsec, которая управляет подключениями StrongSwan:  
Логи хранятся в /var/log/auth.log и /var/log/daemon.log.  
 Данный продукт позволяет на базе linux систем сделать то что нам нужно, но все файл конфигурации будут раскиданы по репозиториям из за чего он нам не подходит.

Следующий кандидат VyoOS: VyOS представляет собой форк известной сетевой операционной системы [Vyatta](http://www.brocade.com/launch/vyatta/). Его первый релиз под кодовым названием Hydrogen был представлен в декабре 2013 года.  
Последний на сегодняшний день релиз — Helium — вышел в свет в сентябре 2014 года. Интерфейс командной строки (CLI) в VyOS похож на СLI устройств от Juniper Networks. Как и Vyatta, VyOS основан на Debian. Это позволяет расширить функциональность путём установки дополнительных deb-пакетов.



Возможности VyOS действительно широки. Вот далеко не полный список:

* межсетевой экран для IPv4 и IPv6, включая фильтрацию p2p-трафика;  
  трансляция сетевых адресов (NAT);
* DHCP-сервер для IPv4 и IPv6;
* система обнаружения вторжений;
* балансировка нагрузки и резервирование канала;
* резервирование маршрутизаторов с синхронизацией таблицы состояний соединений;
* виртуальные частные сети (IPsec, L2TP/IPsec, PPTP, OpenVPN);
* учёт трафика (Netflow и sFlow);
* веб-прокси и фильтрация URL.

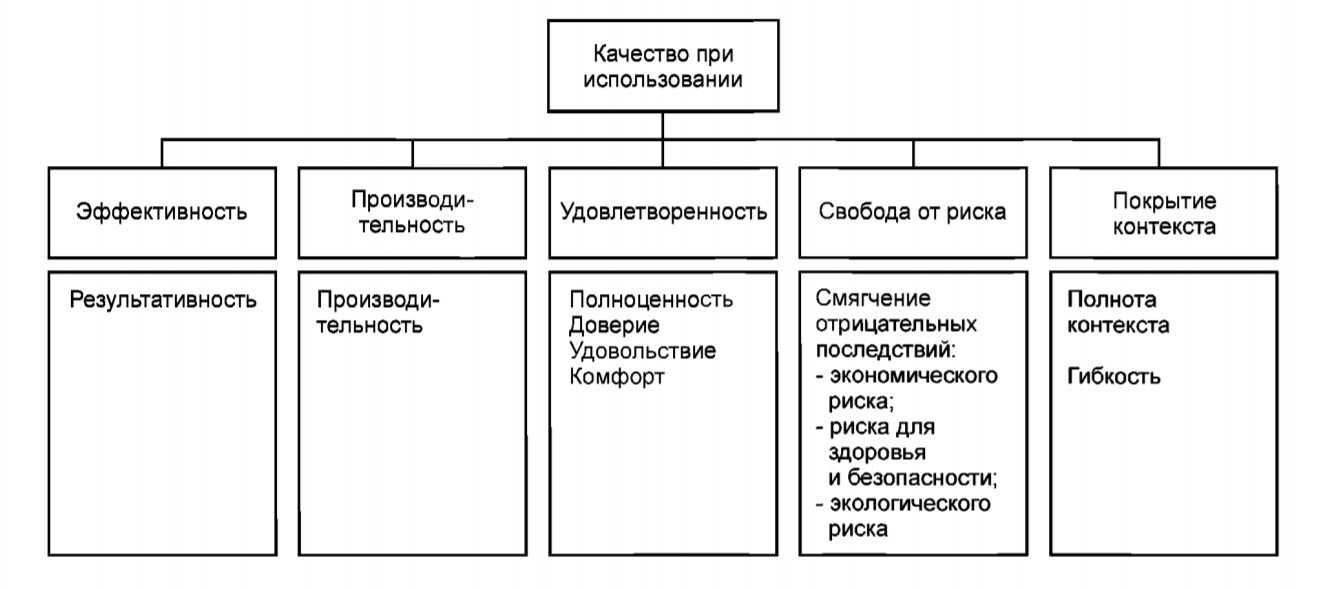
Тем самым данное программное обеспечение нам полностью подходит и у него имеется готовая официальная сборка для разворачивания в среде контейнера.

1.3. Требования автоматизации и отказоустойчивости структуры.

Прежде чем проектировать серверную структуру, которая будет располагаться в облачных сервисах необходимо определить требования и ограничения для нее. В данном проекте планируется, что эта структура будет использоваться для подключений к smsc и ussd серверов, то есть будет ходить большой поток информации. И каждый простой или отказ будет стоить больших убытков. Потому что отсутствующая магистраль до серверов означает, что абоненты не могут получить услуги связи.

Если представить проектирующуюся структуру, как продукт то к нему можно наложить требования ГОСТ Р ИСО/МЭК 25010— 2015 «Требования и оценка качества систем и программного обеспечения». Данный стандарт был выбран по причине отсутствия по данному виду архитектуры других стандартов, ведь проектирование будет осуществляться на уровне выше второго модели OSI. Это связано с тем что коммутацию на уровне 2 и 1 модели OSI осуществляется средствами дата-центра который предоставляет физическое и логическое пространство. Так же данный выбор позволяет не беспокоится о безопасности на уровне 2 и 1 модели OSI, потому что перед выбором дата-центра можно проверить и убедиться в наличии сертификатов безопасности которые будут гарантировать указанный на них уровень безопасности.

Если рассмотреть пункт 3.2 Модель качества при использовании.



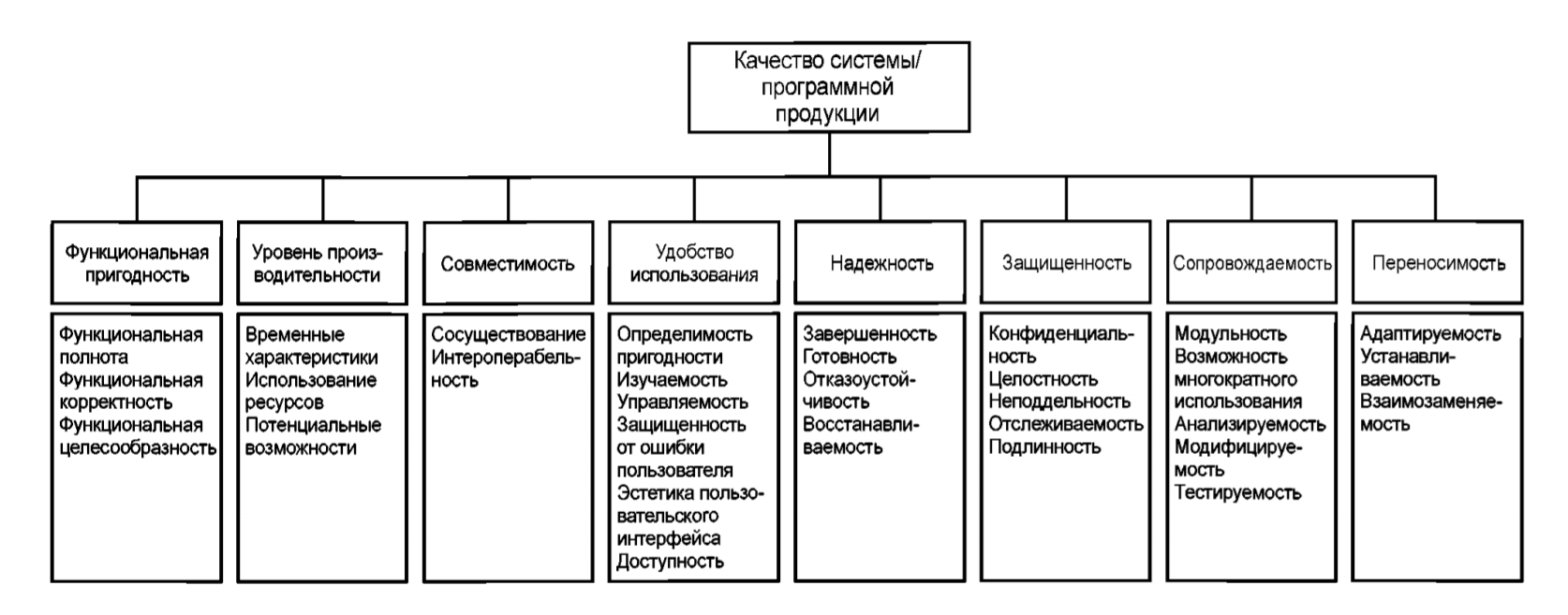
Первые два пункта говорят о том, что система должна в полной мере отвечать поставленным ей задачам. В данном случае это полноценная передача без потерь если рассматривать только эту сторону задачи.

Параметр удовлетворенность размыт так как удовлетворенность будет зависть только от отсутствия с проблем с сетью.

А вот параметр свобода от риска один из самых сложных в данной задаче. Проектируемая Сетевая структура должна обеспечивать резервный маршрут для бесперебойной работы. И в случае выхода из строя по техническим причинам одного контейнера должна использовать другой и делать попытки восстановить первый. Так же должна быть настроена система мониторинга изменений конфигураций, то есть резервное копирование. То есть нужно будет указать в конфигурационном файле количество изменений после которых должна происходить отправка конфигурационного файла на внутренний сервер.

Полноту контекста можно охарактеризовать как: Степень, в которой структура применима при использовании малх вычислительных ресурсов с высокой сетевой пропускной способностью, квалифицированными пользователями и в отказоустойчивом режиме.

В пункте модели качество продукта данного Госта указаны характеристики и под характеристики как показано ниже.



Глава 2. Функциональное моделирование

1.1. Основные характеристики средств программной маршрутизации.