|  |  |
| --- | --- |
|

|  |
| --- |
| МИНОБРНАУКИ РОССИИФедеральное государственное бюджетное образовательное учреждениевысшего образования**«МИРЭА – Российский технологический университет»****РТУ МИРЭА** |

 |

Институт Информационных Технологий

Кафедра корпоративных информационных систем (КИС)

**ОТЧЕТ**

по Практической работе №1

на тему

«Составление плана мероприятий для организации безопасного функционирования ИС на базе ОС Debian 7.11»

по дисциплине

«Безопасность функционирования информационных систем»

Выполнил студент группы ИКБО-08-18 Валяев Д.А.

Принял к.т.н., доцент Башлыкова А.А.

Выполнено «\_\_\_» \_\_\_\_\_\_\_\_\_2021 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_2021 г.

Практическая работа № 1.

Тема: «Составление плана мероприятий для организации безопасного функционирования ИС на базе ОС Debian 7.11»

Цель работы: Анализ выбранной ОС, анализ его ядра, определение категории требований к безопасности РИС.

Раздаточный материал: дидактический материал

Перечень и характеристики оборудования, приборов и материалов:

Тип компьютера: PC

Операционная система: Windows 10 Pro

Пакет обновления: 10.0.18362 Сборка 18362

Internet Explorer: 11.476.18362.0

DirectX: DirectX 12

Тип ЦП: 6х, 3.2 GHz AMD Ryzen 5 1600

Системная память: 16 Гб = 16384 MB

Видеоадаптер: NVIDIA GeForce RTX 2060

Монитор: Samsung Odyssey G3

Дисковый накопитель: Samsung EVO 870 (500 ГБ)

## Исходные данные

##  Топология и составляющие исходной распределенной информационной системы

Распределенная информационная система (РИС) развернута в удаленных друг от друга помещениях. Исходя из того, что РИС обладает большим масштабом и находится в не одном помещении, классифицируем ее как РИС, состоящую из локальных вычислительных сетей (ЛВС), каждая из которых построена на базовой топологии звезда. Топология звезда представляет собой сеть, в которой все компьютеры сети присоединены к центральному узлу (коммутатору), топология звезда представляется, как физический сегмент сети.

Оконечные устройства в ЛВС подключены непосредственно к управляемому коммутатору CISCO серии WS-2960-24tc-l и находятся в одном широковещательном домене. Для физического подключения узлов к сетевому оборудованию используется витая пара UTP 5e. Доступ в интернет обеспечивается через маршрутизатор CISCO серии 3745. Оконечные устройства для получения сетевых настроек используют протокол DHCP. Протокол DHCP сконфигурирован в режиме динамического распределения адресов. Внутренний адрес подсети – 192.168.1.0/24, что соответствует категории С частных адресов. Максимальное количество узлов в данной сети: $2^{32-24}$-2 = 254. Минимальный IP-адрес узла – 192.168.1.1, максимальный IP-адрес – 172.16.1.30, широковещательный (broadcast) 192.168.1.255.

В состав каждой ЛВС входят до 8 автоматизированных рабочих мест (АРМ) и 2 выделенных сервера. Из 8 указанных АРМ в изолированный сегмент сети вынесены 4 АРМ и 1 выделенный сервер. В выделенных АРМ создается, хранится и распространяется конфиденциальная информация. Конфиденциальная информация так же циркулирует в сегменте сети, в котором функционируют АРМ. Сегмент сети, в котором циркулирует конфиденциальная информация, изображен на рисунке 1.



Рис.1.1 Изолированный сегмент сети ЛВС (как части РИС)

АРМ функционируют на базе свободной операционной системы (ОС) Debian 7.11.

 Данная операционная система ОС базируется на ядре Linux версии 3.2.0-4 с симметричной многопроцессорной обработкой (SMP), что обеспечивает поддержку многоядерных и многопроцессорных систем (SMP) и с поддержкой расширения физического адреса (PAE), что позволяет использовать объем оперативной памяти более 4GB с учетом 32-х битной архитектуры.

##  Особенности наличия изолированного сегмента распределенной информационной системы

Корпоративные сети превратились в сложные, высоконагруженные системы. Они состоят из множества серверов различных типов, а также различных рабочих групп, которые нуждаются в связи друг с другом. Если в такой среде не разделить сеть на сегменты, получится хаос, который невозможно контролировать. По мере роста масштаба такой сети надежность и безопасность корпоративной системы будет стремиться к нулю.

Для решения задачи быстрого создания сегмента сети (изолированного сегмента сети) используются виртуальные локальные сети (VLAN). VLAN – группа узлов сети, траффик такой сети изолирован от трафика другой сети на канальном уровне.

Основное назначение технологии VLAN – это облегчение процесса создания изолированного сегмента сети. Такие сегменты сети связываются между собой с помощью коммутатора. Таким образом создается достаточно мощный барьер, препятствующий попаданию нежелательного трафика из одной сети в другую. До появления технологии VLAN для создания сегмента сети использовались физически изолированные сегменты коаксиального кабеля, или же несвязанные между собой элементы. Которые были построены на повторителях и мостах. Такие сети связывались между собой в единую с помощью маршрутизаторов. Этот способ построения сети с изолированным сегментом является достаточно неудобным, затратным. Для его реализации требуется физическое перестроение сети, и использование дополнительно оборудования.

Существует несколько способов создания изолированного сегмента с помощью технологии VLAN:

* С помощью 1 коммутатора;
* С помощью 2 и более коммутаторов.

В первом случае к одному коммутатору подключено несколько хостов, находящихся в различных виртуальных сетях (рис.2).



Рис. 1.2 Две виртуальные сети на одном коммутаторе

После подключения хост-компьютера к порту коммутатора, на коммутатор передается фрейм от хоста, т.е. набор рабочей информации, таким образом в коммутаторе создается таблица коммутации. В простейшем виде она имеет следующий вид (табл. 1.1).

Таблица 1.1 Таблица коммутации

|  |  |
| --- | --- |
| Порт коммутатора | MAC-адрес хоста |
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | D |

В первом столбце таблицы указывается порт коммутатора, а в соответствующем ему втором столбце указывается MAC-адрес хоста (условно A, B, C, D), полученный во фрейме. Как правило, в настройках по умолчанию все порты коммутатора принадлежат одной виртуальной сети – VLAN 1.

В настройках коммутатора можно разделить порты коммутатора, присвоив их к разным виртуальным сетям VLAN1 и VLAN2. Например, первый и второй порт отнести к VLAN 1, а третий и четвертый к VLAN2. В таком случае таблица коммутации будет выглядеть следующим образом (табл. 1.2).

Таблица 1.2 Таблица коммутации для нескольких VLAN

|  |  |  |
| --- | --- | --- |
| Порт коммутатора | VLAN | MAC-адрес хоста |
| 1 | 1 | A |
| 2 | 1 | B |
| 3 | 2 | C |
| 4 | 2 | D |

При использовании двух коммутаторов заполнение таблицы коммутации происходит аналогичным образом, но для того чтобы хосты, подключенные к разным коммутаторам, но отнесенные к одному сегменту сети увидели друг друга требуется соединить и настроить коммутаторы следующим образом (рис. 1.3)



Рис. 1.3 Две виртуальные сети на двух коммутаторах

Свободный порт на коммутаторе 1 и на коммутаторе 2 необходимо добавить в 1 сегмент сети, такой же VLAN, как и хосты, требующие объединения в 1 сегмент, и соединить эти порты между собой.

По организации работы VLAN можно сделать вывод, что изолированный сегмент в РИС обладает рядом достоинств:

* Увеличение безопасности в сети;
* Позволяет осуществить гибкое деление устройств на группы.

**ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ:**

1. **Выбор ОС для работы в рассматриваемой распределенной информационной системе.**

Для рассматриваемой РИС была выбрана ОС Debian 7.11 (wheezy), обусловлено это тем, что дистрибутивы Debian более стабильны, чем Ubuntu, а ядро Linux 3.2.0-4 используется только в Debian 7 и Ubuntu 12.04.

Операционная система Debian 7 имеет статус LTS (Long Time Support) – версии. Кодовое название Debian 7 – Wheezy. Первая версия вышла 04.06.2016, и поддержка продолжалась до конца мая 2018 года. Начиная с версии 7, в Debian появилась возможность запуска как 64 битных программ, так и 32 битных на операционных системах с архитектурой х64, благодаря технологии MultiArch (мультиархитектурность) [1].

Минимальные системные требования Debian 7:

* Процессор с тактовой частотой от 700 МГц;
* 3 Гб свободного места на жестком диске;
* Объем оперативной памяти от 256 Мб;
* Наличие DVD/CD привода или порта USB [2].

Одним из главных плюсов Debian является интероперабельность, что положительно сказывается на количестве и качестве совместимого ПО.

Особенности обеспечения безопасности в Linux 12.04:

* Системный пароль, который используется для входа в систему, может быть сохранен в файле /etc/shadow (доступ к которому имеет только пользователь ROOT) и хеширован с использованием алгоритмов MD5, SHA-256 или SHA-512 и соли (salt), что значительно усложняет подбор пароля;
* В системе присутствует возможность шифрования файловой системы посредством Cryptomator;
* По умолчанию в системе нет открытых портов;
* Возможность добавления пользователей, а также групп пользователей с разными правами доступа [3].
1. **Описание основных характеристик ядра выбранной ОС**
	1. **Особенности ядра Debian 7.11:**

В планировщик задач добавлена возможность ограничения времени потребления CPU для группы процессов, например, если система не нагружена, подобная возможность позволит выделить больше ресурсов процессам, которые в иной ситуации были бы ограничены более жёстко;

* Увеличение отзывчивости десктоп-приложений в условиях большой интенсивности операций записи данных;
* Новые системные вызовы прямого обращения к областям памяти других процессов, позволяют избегать копирования данных, что ускоряет взаимодействие приложений;
* В TCP-стек интегрирована поддержка алгоритма пропорционального снижения интенсивности отправки пакетов;
* Увеличение скорости выполнения операции проверки целостности данных. Использование упреждающего чтения позволило сократить время проверки тестового раздела и увеличить пропускную способность с 70 до 90%;
* Улучшенная реализация алгоритма хеширования SHA1, который оптимизирован для процессоров x86-64;
* Добавлен API для настройки параметров подсистемы crypto из пространства пользователя (например, можно выбирать используемые алгоритмы шифрования) [4].
	1. **Добавление пакетов в репозиторий**

Добавление пакетов в репозиторий осуществляется с помощью встроенной системы управления пакетами, доступ к которой происходит через консоль. Для добавления репозитория используется команда «sudo apt-add-repository 'deb package name'», удаление репозитория производится командой «sudo apt-add-repository --remove 'deb package name'» [13].

* 1. **Поддержание архитектур**

Начиная с версии 7, в Debian появилась возможность запуска как 64 битных программ, так и 32 битных на ОС с архитектурой х64, благодаря технологии MultiArch (мультиархитектурность) [1].

* 1. **Использование периферийных устройств**

При подключении периферийных устройств Debian, в большинстве случаев определяет их автоматически, если автоматического определения не произошло, можно перейти в параметры загрузки, чтобы указать присутствующие периферийные устройства [6].

* 1. **Работа с буфером обмена**

В Debian, как и в других ОС, работающих на ядре Linux, есть два независимых друг от друга буфера обмена:

* Clipboard, который работает также, как и буфер обмена в ОС Windows: Ctrl+C для копирования элементов и Ctrl+V для вставки элементов из буфера обмена;
* Primary, который работает совершенно иначе: при выделении текста мышью, он сразу попадает в буфер обмена, чтобы вставить текст необходимы нажать на колесико или одновременно нажать правую и левую кнопки мыши.

Среди сторонних программ для работы с буфером обмена можно выделить две:

* Parcellite - менеджер буфера обмена на GTK для среды GNOME
* klipper - менеджер буфера обмена для среды KDE

Кроме программ для работы с буфером обмена, имеющих графический интерфейс, есть консольные программы xclip и xsel, которые позволяют управлять буфером обмена с помощью простых команд, что дает возможность использовать их в скриптах [7].

* 1. **Улучшения совместимости в РИС**

При работе с распределенной информационной системой, логично использование git, и тут следует заметить простоту работы с git-репозиториями в Debian. Установка git производится посредством выполнения команды «sudo apt install git», а синтаксис команд взаимодействия с репозиториями очень прост для понимания, что упрощает пользователям обмен файлмами.

* 1. **Добавление драйверов**

В Debian ситуация с драйверами сильно отличается от Windows. Большинство драйверов для периферийного оборудования распространяются в виде открытого программного обеспечения и интегрированы в дистрибутив Debian. Эти драйверы, как правило, являются частью ядра Linux, но здесь есть исключение - драйверы видеокарты являются частью программного обеспечения Xorg, а драйверы принтера включены в набор CUPS (система печати).

Это значит, что большинство доступных для linux драйверов уже установлены на вашем компьютере, включены вместе с ядром, в виде модулей ядра или вместе с графическим сервером или сервером печати. Установка драйверов на системы с ядром Linux почти не нужна. Драйверы в Linux разработаны, в основном, энтузиастами, но иногда их выпускают сами производители оборудования, которые вносят свой вклад в развитие ядра, например, это Intel, AMD, а также другие, более мелкие производители.

Для ручной установки необходимого драйвера требуется ввести команду «sudo apt-add-repository repository\_name» и получить необходимые обновления командой «sudo apt-get update» и «sudo apt-get dist-upgrade» [8].

* 1. **Возможность пересборки установщика системы**

Linux — одна из немногих операционных систем, позволяющих воздействовать на работу компьютера, модифицируя ядро системы. Поскольку Linux поставляется с полным набором исходного кода, в нем можно опробовать любую новинку. Конечно, большинство предпочтет модифицировать ядро при помощи инструментов, входящих в комплект, а не компилировать код, написанный на языке С. Тем не менее эта ОС имеет открытый исходный код, что позволяет при большом желании модернизировать ее и подстроить под свои специфические нужды [5].

* 1. **Экранирование HTML элементов системы**

Перед запуском введенной команды интерпретатор анализирует ее на наличие так метасимволов, т.е. символов, имеющих специальное значение. Если в команде есть метасимвол, который не должен быть воспринят как часть команды, его необходимо экранировать, в Debian это делается с помощью спецсимвола «\». В тех случаях, когда там нужен символ «\», но при этом он не должен выполнять экранирование, используется конструкция «\\». Для экранирования последовательности символов (текста) необходимо заключить его в кавычки, что отменит действие всех спецсимволов, кроме «/» и «$». В случае, когда надо экранировать все символы последовательности, в том числе и «/» и «$», необходимо использовать одинарные кавычки [10].

* 1. **Решение проблем совместимости**

В начале развития Linix, поскольку большая часть ПО создается не большими компаниями, а энтузиастами, была проблема с совместимостью потому, что ПО создавалось под какой-то конкретный дистрибутив, определенной версии и не было никаких гарантий, что оно будет работать на следующей версии этого же дистрибутива. После введения разработчиками стандартов создания ПО, проблемы с совместимостью были решены.

Сегодня ПО создается в расчете на использование в абстрактном Linux, а не в определенном дистрибутиве, поэтому задача совместимости перекладывается на сборщиков дистрибутивов. Проблема совместимости дистрибутивов и прикладных программ, существует и сейчас, но касается она не открытого ПО, а недоступного в исходном коде, следовательно, его нельзя адаптировать под другую систему.

1. **Определение класса защищенности**

Класс защищенности данной РИС – 2.

Требования, предъявляемые к системе:

* Разработка политики идентификации и аутентификации;
* Идентификация и аутентификация пользователей и инициируемых ими процессов;
* Идентификация и аутентификация устройств;
* Управление идентификаторами;
* Управление средствами аутентификации;
* Идентификация и аутентификация внешних пользователей;
* Защита аутентификационной информации при передаче;
* Разработка политики управления доступом;
* Управление учетными записями пользователей;
* Реализация политик управления доступа;
* Доверенная загрузка;
* Разделение полномочий (ролей) пользователей;
* Назначение минимально необходимых прав и привилегий;
* Ограничение неуспешных попыток доступа в информационную (автоматизированную) систему;
* Блокирование сеанса доступа пользователя при неактивности;
* Управление действиями пользователей до идентификации и аутентификации;
* Реализация защищенного удаленного доступа;
* Контроль доступа из внешних информационных (автоматизированных) систем;
* Разработка политики ограничения программной среды;
* Управление установкой (инсталляцией) компонентов программного обеспечения;
* Разработка политики защиты машинных носителей информации;
* Учет машинных носителей информации;
* Управление физическим доступом к машинным носителям информации;
* Контроль использования интерфейсов ввода (вывода) информации на машинные носители информации;
* Контроль подключения машинных носителей информации;
* Уничтожение (стирание) информации на машинных носителях информации;
* Разработка политики аудита безопасности;
* Инвентаризация информационных ресурсов;
* Анализ уязвимостей и их устранение;
* Генерирование временных меток и (или) синхронизация системного времени;
* Регистрация событий безопасности;
* Защита информации о событиях безопасности;
* Мониторинг безопасности;
* Реагирование на сбои при регистрации событий безопасности;
* Проведение внутренних аудитов;
* Разработка политики антивирусной защиты;
* Реализация антивирусной защиты;
* Антивирусная защита электронной почты и иных сервисов;
* Обновление базы данных признаков вредоносных компьютерных программ (вирусов);
* Разработка политики предотвращения вторжений (компьютерных атак);
* Обнаружение и предотвращение компьютерных атак;
* Обновление базы решающих правил;
* Разработка политики обеспечения целостности;
* Контроль целостности программного обеспечения;
* Контроль данных, вводимых в информационную (автоматизированную) систему;
* Контроль ошибочных действий пользователей по вводу и (или) передаче информации и предупреждение пользователей об ошибочных действиях;
* Разработка политики обеспечения доступности;
* Использование отказоустойчивых технических средств;
* Резервирование средств и систем;
* Контроль безотказного функционирования средств и систем;
* Резервное копирование информации;
* Обеспечение возможности восстановления информации;
* Обеспечение возможности восстановления программного обеспечения при нештатных ситуациях;
* Контроль предоставляемых вычислительных ресурсов и каналов связи;
* Разработка политики защиты технических средств и систем;
* Организация контролируемой зоны;
* Управление физическим доступом;
* Размещение устройств вывода (отображения) информации, исключающее ее несанкционированный просмотр;
* Защита от внешних воздействий;
* Разработка политики защиты информационной (автоматизированной) системы и ее компонентов;
* Разделение функций по управлению (администрированию) информационной (автоматизированной) системой с иными функциями;
* Защита периметра информационной (автоматизированной) системы;
* Эшелонированная защита информационной (автоматизированной) системы;
* Сегментирование информационной (автоматизированной) системы;
* Организация демилитаризованной зоны;
* Сокрытие архитектуры и конфигурации информационной (автоматизированной) системы;
* Использование программного обеспечения, функционирующего в средах различных операционных систем;
* Защита неизменяемых данных;
* Защита от спама;
* Защита информации при ее передаче по каналам связи;
* Обеспечение доверенных канала, маршрута;
* Запрет несанкционированной удаленной активации периферийных устройств;
* Контроль использования мобильного кода;
* Контроль передачи речевой информации;
* Контроль передачи видеоинформации;
* Обеспечение подлинности сетевых соединений;
* Исключение возможности отрицания отправки информации;
* Исключение возможности отрицания получения информации;
* Защита беспроводных соединений;
* Защита от угроз отказа в обслуживании (DOS, DDOS-атак);
* Управление сетевыми соединениями;
* Защита информации при использовании мобильных устройств;
* Управление перемещением виртуальных машин (контейнеров) и обрабатываемых на них данных;
* Разработка политики реагирования на компьютерные инциденты;
* Выявление компьютерных инцидентов;
* Информирование о компьютерных инцидентах;
* Анализ компьютерных инцидентов;
* Устранение последствий компьютерных инцидентов;
* Принятие мер по предотвращению повторного возникновения компьютерных инцидентов;
* Разработка политики управления конфигурацией информационной (автоматизированной) системы;
* Управление изменениями;
* Установка (инсталляция) только разрешенного к использованию программного обеспечения;
* Разработка политики управления обновлениями программного обеспечения;
* Поиск, получение обновлений программного обеспечения от доверенного источника;
* Контроль целостности обновлений программного обеспечения;
* Тестирование обновлений программного обеспечения;
* Установка обновлений программного обеспечения;
* Разработка политики планирования мероприятий по обеспечению защиты информации;
* Разработка, утверждение и актуализация плана мероприятий по обеспечению защиты информации;
* Контроль выполнения мероприятий по обеспечению защиты информации;
* Разработка политики обеспечения действий в нештатных ситуациях;
* Разработка плана действий в нештатных ситуациях;
* Обучение и отработка действий персонала в нештатных ситуациях;
* Создание альтернативных мест хранения и обработки информации на случай возникновения нештатных ситуаций;
* Резервирование программного обеспечения, технических средств, каналов связи на случай возникновения нештатных ситуаций;
* Обеспечение возможности восстановления информационной (автоматизированной) системы в случае возникновения нештатных ситуаций;
* Анализ возникших нештатных ситуаций и принятие мер по недопущению их повторного возникновения;
* Разработка политики информирования и обучения персонала;
* Информирование персонала об угрозах безопасности информации и о правилах безопасной работы;
* Обучение персонала правилам безопасной работы;
* Проведение практических занятий с персоналом по правилам безопасной работы;
* Контроль осведомленности персонала об угрозах безопасности информации и о правилах безопасной работы [12].
1. **Действия и работы для обеспечения централизованного хранения критически важных резервных копий системы**

Для обеспечения централизованного хранения критически важных резервных копий системы должна использоваться система TimeMachine. Поддержка TimeMachine появилась в Samba версии 4.8, а Debian 7 изначально не поддерживает данную версию, поэтому Samba придется собирать вручную.

После установки Samba на сервер под управлением Debian, необходимо произвести изначальную настройку, куда входит:

* настройка сети;
* отключение всех неиспользуемых служб;
* установка обновления пакетов;
* установить NTP-клиент;
* установить сетевой протокол Kerberos.

После выполнения первоначальной настройки сервера необходимо настроить домен в NTP для возможности последующего обращения к серверу. Чтобы идентифицировать пользователей используется сетевой протокол Kerberos. Последним шагом настройки системы является разметка диска для хранения резервных копий.

Для безопасности сервера необходимо ограничить вход в систему определенным группам пользователей [11].

Вывод: В настоящее время существует довольно большое множество операционных систем, отвечающих самым разносторонним требованием пользователей имеющих различную степень открытости. Для данной распределенной информационной системы выбрана ОС Debian 7, потому что она имеет увеличенный срок поддержки, отвечает требованиям открытости и защищенности, проста в установке и эксплуатации.

Безопасность функционирования данной РИС должна обеспечиваться за счет использования как стандартных средств защиты ОС, таких как ограничение прав доступа к данным, ограничение доступа к неиспользуемым портам, так и сторонних механизмов организации безопасного функционирования системы, таких как настройка своевременного создания резервных копий важных данных, использование средств идентификации пользователей при работе с общими данными.

Список источников

1. Информация о выпуске Debian “wheezy” // [Электронный ресурс] – URL: https://www.debian.org/releases/wheezy/.
2. Минимальные системные требования // [Электронный ресурс] – URL: https://systemrequirements.ru/linux-debian-7-sistemnyie-trebovaniya/
3. Информация по безопасности // [Электронный ресурс] – URL: <https://www.debian.org/security/>
4. Релиз ядра Linux 3.2. Обзор новшеств // [Электронный ресурс] – URL: <https://www.opennet.ru/opennews/art.shtml?num=32727>
5. Рекомпиляция ядра Linux // [Электронный ресурс] – URL: <http://samoychiteli.ru/document32796.html>
6. Параметры загрузки – Debian // [Электронный ресурс] – URL: <https://www.debian.org/releases/wheezy/mips/ch05s03.html.ru>
7. Буфер обмена Linux // [Электронный ресурс] – URL: <http://mydebianblog.blogspot.com/2011/01/linux.html>
8. Установка драйверов в Linux // [Электронный ресурс] – URL: <https://losst.ru/ustanovka-drajverov-v-linux>
9. Совместимость Linux: есть ли проблема? / Алексей Федорчук // [Электронный ресурс] – URL: <https://www.alv.me/sovmestimost-linux-est-li-problema/>
10. Спецсимволы в bash // [Электронный ресурс] – URL: <https://parallel.uran.ru/book/export/html/461>
11. Организация сетевого ресурса для Time Machine на Debian // [Электронный ресурс] – URL: <https://blog.it-kb.ru/2019/06/11/time-machine-backup-server-for-macos-clients-with-centralized-smb-storage-on-macos-linux-debian-10-buster-or-windows-server-2012-r2/>
12. Приказ ФСТЭК России от 14 марта 2014 г. N 31 // [Электронный ресурс] – URL: <https://fstec.ru/normotvorcheskaya/akty/53-prikazy/868-prikaz-fstek-rossii-ot-14-marta-2014-g-n-31>
13. Как добавить репозиторий в Debian // [Электронный ресурс] – URL: <https://losst.ru/kak-dobavit-repozitorij-v-debian>