Краснодарский университет МВД России

АДМИНИСТРИРОВАНИЕ ПОДСИСТЕМ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Учебное пособие

Краснодар
2021

УДК 004.056+004.383.2
ББК 16.2+16.8
А313

Одобрено
редакционно-издательским советом
Краснодарского университета
МВД России

Составители: *М. Ю. Макуха, Д. С. Богданов*

Рецензенты:
А. И. Примакин, доктор технических наук, профессор (Санкт-Петербургский университет МВД России);
Д. Н. Саратов, кандидат технических наук, доцент (Санкт-Петербургский университет МВД России).

А313 **Администрирование подсистем** обеспечения информационной безопасности [Электронный ресурс] : учебное пособие / сост.: М. Ю. Макуха, Д. С. Богданов. – Электрон. дан. – Краснодар : Краснодарский университет МВД России, 2021. – 1 электрон. опт. диск.

ISBN 978-5-9266-1731-0

Рассматриваются основные средства и способы обеспечения информационной безопасности в автоматизированных системах. Отражены концепции и принципы администрирования подсистем обеспечения информационной безопасности.

Для профессорско-преподавательского состава, адъюнктов, курсантов, слушателей образовательных организаций МВД России и сотрудников органов внутренних дел Российской Федерации.

УДК 004.056+004.383.2
ББК 16.2+16.8

ISBN 978-5-9266-1731-0

ОГЛАВЛ ЕНИЕ

ВВЕДЕНИЕ

Процесс обеспечения безопасности относится к оперативным процессам и, в соответствии с библиотекой ITIL, входит в блок процессов поддержки ИТ-сервисов. Нарушение безопасности информационной системы предприятия может привести к ряду негативных последствий, влияющих на уровень предоставления ИТ-сервисов:

– снижение уровня доступности вследствие отсутствия доступа или низкой скорости доступа к данным, приложениям или службам;

– полная или частичная потеря данных;

– несанкционированная модификация данных;

– получение доступа посторонних пользователей к конфиденциальной информации.

Анализ причин нарушения информационной безопасности показывает, что основными являются следующие:

– ошибки конфигурирования программных и аппаратных средств ИС;

– случайные или умышленные действия конечных пользователей и сотрудников ИТ-службы;

– сбои в работе программного и аппаратного обеспечения ИС;

– злоумышленные действия посторонних по отношению к информационной системе лиц.

Концепция защищенных компьютерных систем построена на четырех принципах:

– безопасность, которая предполагает создание максимально защищенных ИТ-инфраструктур;

– конфиденциальность, которая подразумевает внедрение в состав и технологий и продуктов средств защиты конфиденциальности на протяжении всего периода их эксплуатации;

– надежность, которая требует повышения уровня надежности процессов и технологий разработки программного обеспечения информационных систем;

– целостность деловых подходов для укрепления доверия клиентов, партнеров, государственных учреждений.

Для решения вопросов обеспечения информационной безопасности компания Microsoft предоставляет следующие технологии:

– Active Directory – единый каталог, позволяющий сократить число паролей, которые должен вводить пользователь;

– двухэтапная аутентификация на основе открытых/закрытых ключей и смарт-карт;

– шифрование трафика на базе встроенных средств операционной системы IPSec;

– создание защищенных беспроводных сетей;

– шифрование файловой системы;

– защита от вредоносного кода;

– организация безопасного доступа мобильных и удаленных пользователей;

– защита данных на основе кластеризации, резервного копирования и ограничения несанкционированного доступа;

– служба сбора событий из системных журналов безопасности.

Разработка учебного пособия обусловлена необходимостью разработки методического обеспечения дисциплины «Администрирование подсистем обеспечения информационной безопасности» по специальности 10.05.05 – Безопасность информационных технологий в правоохранительной сфере.

Учебное пособие позволит получить знания и навыки по администрированию элементов обеспечения информационной безопасности рабочих станций и серверов работающих под управлением операционных систем Linux и Windows.

1. ОБЩИЕ ПОЛОЖЕНИЯ АДМИНИСТРИРОВАНИЯ ПОДСИСТЕМ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

1.1. Основные понятия, концепции и принципы администрирования подсистем обеспечения информационной безопасности

Информация, как объект защиты, имеет ряд особенностей:

она нематериальна, т.к. невозможно измерить известными физическими приборами ее массу, размеры, энергию;

может храниться, обрабатываться, передаваться по каналам связи в виде сообщений, записанных на различные материальные носители.

Сущность семантической информации не зависит от характеристик носителя. Содержание текста, например, не зависит от качества бумаги, на которой он написан, или физических параметров другого носителя. Семантическая информация - продукт абстрактного мышления человека, отображаемые с помощью символов (сообщений) на языках общения людей объекты, явления, образы и модели. Языки общения включают как естественные языки национального общения, так искусственные профессиональные языки.

Профессиональные языки создаются специалистами для экономного и компактного отображения информации. Существует множество профессиональных языков: математики, музыки, радиоэлектроники, автодорожного движения, химии и т.д. Любая предметная область содержит характерные для нее понятия и условные обозначения, часто непонятные необученному этому языку человеку.

Информация признаковая описывает конкретный материальный объект на языке его признаков. Описание объекта содержит признаки его внешнего вида, излучаемых им полей и элементарных частиц, состава и структуры веществ, из которых состоит объект. Источниками признаковой информации являются сами объекты.

Носителями информации являются материальные объекты, обеспечивающие запись, хранение и передачу информации в пространстве и

времени. Носителями информации являются: люди, макротела, поля, микрочастицы (элементарные частицы).

Макротела (материальные тела) являются наиболее долговременными носителями различных видов информации. Прежде всего, материальные тела содержат информацию о своем составе, структуре (строении), о воздействии на них других материальных тел., например, по остаточным изменениям структуры бумаги восстанавливают подчищенные надписи, по изменению структуры металла двигателя определяют его заводской номер, перебитый автомобильными ворами. Материальные тела (папирус, глиняные таблички, береста, камень, бумага) использовались людьми для консервации и хранения информации в течение всей истории человечества. И в настоящее время бумага является самым распространенным носителем семантической информации. Однако четко прослеживается тенденция замены бумаги машинными носителями (магнитными, полупроводниковыми, светочувствительными и др.

Носителями информации являются также различные поля. Из известных полей в качестве носителей применяются акустические, электрические, магнитные и электромагнитные (в радиодиапазоне, диапазоне видимого и инфракрасного света). Информация содержится в значениях параметров полей. Если поля представляют собой волны, то информация содержится в их амплитуде, частоте и фазе.

Из многочисленных элементарных частиц в качестве носителей информации используются электроны, образующие статические заряды и электрический ток, а также частицы (электроны и ядра гелия) радиоактивных излучений.

Человек как носитель информации ее запоминает и пересказывает получателю в письменном виде или устно. При этом он может полученную от источника информацию преобразовать в соответствии с собственным толкованием ее содержания, исказив смысл.

Различают носители - источники информации, носители - переносчики информации, носители - получатели информации. Например, чертеж является

источником информации, а бумага, на которой он нарисован, - носитель информации. Физическая природа источника и носителя в этом примере одна и та же бумага. Однако между ними существует разница. Бумага, без нанесенного на ней текста или рисунка, может быть источником информации о ее физических и химических свойствах. Если бумага содержит семантическую информацию, ей присваивается другое имя: чертеж, документ и т.д. Чертеж детали или узла входит в состав более сложного документа - чертежа прибора, механизма или машины и т.д. вплоть до конструкторской документации образца продукции.

Другие носители, например, поля не имеют четких границ в пространстве, но в любом случае их характеристики измеряемы. Физическая природа носителя-источника информации, носителя-переносчика и носителя-получателя может быть, как одинаковой, так и разной.

1.1.1. Основные термины и определения

1. информация - сведения (сообщения, данные) независимо от формы их представления;

2. информационные технологии - процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов;

3. информационная система - совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств;

4. информационно-телекоммуникационная сеть - технологическая система, предназначенная для передачи по линиям связи информации, доступ к которой осуществляется с использованием средств вычислительной техники;

5. обладатель информации - лицо, самостоятельно создавшее информацию либо получившее на основании закона или договора право разрешать или ограничивать доступ к информации, определяемой по каким-либо признакам;

6. доступ к информации - возможность получения информации и ее использования;

7. конфиденциальность информации - обязательное для выполнения лицом, получившим доступ к определенной информации, требование не передавать такую информацию третьим лицам без согласия ее обладателя;

8. предоставление информации - действия, направленные на получение информации определенным кругом лиц или передачу информации определенному кругу лиц;

9. распространение информации - действия, направленные на получение информации неопределенным кругом лиц или передачу информации неопределенному кругу лиц;

10. электронное сообщение - информация, переданная или полученная пользователем информационно-телекоммуникационной сети;

11. документированная информация - зафиксированная на материальном носителе путем документирования информация с реквизитами, позволяющими определить такую информацию или в установленных законодательством Российской Федерации случаях ее материальный носитель.

12. электронный документ - документированная информация, представленная в электронной форме, то есть в виде, пригодном для восприятия человеком с использованием электронных вычислительных машин, а также для передачи по информационно-телекоммуникационным сетям или обработки в информационных системах;

13. оператор информационной системы - гражданин или юридическое лицо, осуществляющие деятельность по эксплуатации информационной системы, в том числе по обработке информации, содержащейся в ее базах данных;

14. сайт в сети "Интернет" - совокупность программ для электронных вычислительных машин и иной информации, содержащейся в информационной системе, доступ к которой обеспечивается посредством информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет") по доменным

именам и (или) по сетевым адресам, позволяющим идентифицировать сайты в сети "Интернет";

15. страница сайта в сети "Интернет" (далее также - интернет-страница) - часть сайта в сети "Интернет", доступ к которой осуществляется по указателю, состоящему из доменного имени и символов, определенных владельцем сайта в сети "Интернет";

16. доменное имя - обозначение символами, предназначенное для адресации сайтов в сети "Интернет" в целях обеспечения доступа к информации, размещенной в сети "Интернет";

17. сетевой адрес - идентификатор в сети передачи данных, определяющий при оказании телематических услуг связи абонентский терминал или иные средства связи, входящие в информационную систему;

18. владелец сайта в сети "Интернет" - лицо, самостоятельно и по своему усмотрению определяющее порядок использования сайта в сети "Интернет", в том числе порядок размещения информации на таком сайте;

19. провайдер хостинга - лицо, оказывающее услуги по предоставлению вычислительной мощности для размещения информации в информационной системе, постоянно подключенной к сети "Интернет";

20. единая система идентификации и аутентификации - федеральная государственная информационная система, порядок использования которой устанавливается Правительством Российской Федерации, и которая обеспечивает в случаях, предусмотренных законодательством Российской Федерации, санкционированный доступ к информации, содержащейся в информационных системах;

21. поисковая система - информационная система, осуществляющая по запросу пользователя поиск в сети "Интернет" информации определенного содержания и предоставляющая пользователю сведения об указателе страницы сайта в сети "Интернет" для доступа к запрашиваемой информации, расположенной на сайтах в сети "Интернет", принадлежащих иным лицам, за исключением информационных систем, используемых для осуществления

государственных и муниципальных функций, оказания государственных и муниципальных услуг, а также для осуществления иных публичных полномочий, установленных федеральными законами.

Территориальные сегменты Интегрированной мультисервисной телекоммуникационной системы органов внутренних дел как типовые инфокоммуникационые системы Территориальные сегменты Интегрированной мультисервисной телекоммуникационной системы МВД являются типовыми ИКС ОВД.

Это связано с их относительной автономностью, аккумулированием существенной доли всей информации МВД, а также ролью как систем обеспечения основного исполнительного звена в системе управления ОВД. Как уже было отмечено выше, территориальными сегментами

Интегрированной мультисервисной телекоммуникационной системы (ИМТС) ОВД являются компьютерные системы территориальных органов — министерств внутренних дел субъектов Российской Федерации, главных управлений (управлений) внутренних дел.

Данные системы как инструмент информационной поддержки деятельности этих органов обеспечивают решение следующих задач:

– управление подразделениями полиции, постами и нарядами при оперативном реагировании на поступающие в дежурные части горрайорганов сообщения и заявления граждан о происшествиях и преступлениях;

– обработка и получение необходимых сведений в кратчайшие сроки;

– выявление информационных связей между лицами, событиями и предметами за счет использования интегрированных банков данных, функционирующих в интересах различных служб и позволяющих накапливать, комплексно обрабатывать и получать информацию об объектах специальных, оперативно-разыскных, профилактических и справочных учетов при строгом соблюдении правил доступа к сведениям;

– получение аналитико-статических сведений в различных разрезах и по любой совокупности фиксируемых в учетных документах признаков.

Технологическую основу территориального сегмента ИМТС составляет информационно-вычислительная сеть (ИВС). Как правило, ИВС строятся по иерархическому принципу, где каждый уровень может иметь произвольную топологию: от линейной до звездообразной, включая внутреннюю иерархию архитектуры. При этом каждый уровень представляется одной или несколькими локально-вычислительными сетями (ЛВС), соединенными при помощи каналов удаленного доступа, созданных аппаратурой каналообразования.

Абонентский пункт (АП, в некоторых источниках, в частности — хост (host) — сетевой компьютер.

Маршрутизатор (М, в некоторых источниках, в частности, — роутер (router) — устройство, обеспечивающее маршрутизацию информационных пакетов в сети.

Локальная вычислительная сеть (ЛВС, в некоторых источниках, в частности, — подсеть (subnetwork) — логическое объединение, совокупность абонентских пунктов, являющихся частью ИВС, для которых маршрутизатором выделен одинаковый номер. Абонентские пункты внутри одной ЛВС могут взаимодействовать непосредственно, минуя маршрутизатор.

Физический сегмент ЛВС — физическое объединение абонентских пунктов, образующихся, например, совокупностью АП, подключенных к серверу по схеме «общая шина». При такой схеме подключения каждый абонентский пункт имеет возможность подвергать анализу любой пакет в своем физическом сегменте.

Территориальные сегменты ИМТС представляют собой совокупность объектов, связанных между собой через сетевые узлы. При этом под маршрутом понимается последовательность узлов

сети, по которой данные передаются от источника к приемнику, а под маршрутизацией — выбор маршрута. Все маршрутизаторы имеют специальную таблицу, называемую таблицей маршрутизации, в которой для каждого адресата указывается оптимальная последовательность узлов.

Основные понятия Информационно-вычислительная система (ИВС) — комплекс программных и аппаратных средств для обеспечения автоматизации производства и других сфер жизнедеятельности человека, включающий в качестве составных частей серверное и сетеобразующее оборудование. Пользователь ИВС (User) - физическое лицо, имеющее доступ к определенным ресурсам ИВС. идентифицируемое бюджетом пользователя (учетной записью). Администратор ИВС также является пользователем ИВС. обладая, в общем случае, неограниченным доступом ко всем ресурсам ИВС.

Основные понятия Администратор ИВС (Administrator) - должностное лицо, ответственное за работоспособность и надлежащее функционирование всех частей ИВС. У администратора большой ИВС в подчинении могут находиться администраторы частей и подсистем ИВС, например администратор локально-вычислительной сети, администратор сетевой ОС, администратор БД, а также технический персонал. Администратор подсистемы ИВС отвечает за работоспособность и надлежащее функционирование вверенных ему компонентов этой подсистемы ИВС.

Основные понятия Учетная запись пользователя (Account) — запись в специализированной БД (БД учетных записей), содержащая информацию о пользователе ИВС. Используется для идентификации пользователя в системе, проверки полномочий пользователя и обеспечения доступа пользователя к тем или иным ресурсам системы. Характеризуется атрибутами, например имя для входа, пароль, профиль в системе, список принадлежности к группам и т. п. Пароль служит для защиты бюджета от несанкционированного использования. Регистрация пользователя в системе (Registration) — создание администратором ИВС (или другим уполномоченным лицом) бюджета пользователя данного физического липа.

Основные понятия Аутентификация в системе (Authentication) — процесс установления подлинности пользователя ИВС. Заключается в предъявлении пользователем своего имени для входа и пароля, а также в проверке системой наличия бюджета в БД бюджетов пользователей и соответствия указанного

пользователем пароля и пароля, хранящегося в БД. После успешной аутентификации в системе для пользователя на время сеанса работы создается дескриптор безопасности, отражающий его цифровой идентификатор в системе, а также принадлежность группам пользователей, профилям и другим объектам системы безопасности.

Основные понятия Ресурсы ИВС (Resources) - физические и логические объекты ИВС, имеющие определенную функциональность, доступную для использования. Примеры физических ресурсов - сервер ЛВС, каталог совместного использования на сервере, сетевой принтер; логических ресурсов пользователь, группа пользователей, профиль в системе, очередь на печать и т.д. Совместное использование ресурса (Resource sharing) - использование ресурса двумя и более пользователями ИВС.

Основные понятия Права доступа к ресурсу (Access rights to the resource) - степень свободы действий пользователя (просмотр, использование, владение) по отношению к данному ресурсу. Имеют специфику применительно к разным ресурсам и подсистемам ИВС (создание, чтение, запись, удаление файлов и каталогов - для файловой службы; создание, печать документов/управление очередью на печать — для службы печати и т. д.). По определению, администратор ИВС имеет полные права на все ресурсы ИВС. Администратор части ИВС - полные права на ресурсы части ИВС. Права доступа на прямую связаны с ответственностью пользователя, которую он несет, пользуясь этими правами.

Основные понятия Назначение прав доступа к ресурсу (User's rights assignment) — процедура создания в системе специальной записи, с помощью которой учетной записи пользователя или ее аналогу (например, учетной записи группы пользователей) присваиваются определенные права доступа к ресурсу. Назначение прав доступа в современных информационно-вычислительных системах осуществляется через списки управления доступом (Access Control List/ACL). Аудит /Контроль использования ресурсов (Audit) процесс контроля использования ресурсов, включающий возможность ведения количества

попыток доступа к ресурсам. Журнал аудита ведется на основе данных, поступающих от процедур авторизации.

Основные понятия Список управления доступом (Access Control List /ACL) - в виде отдельных записей хранит информацию о том, кто обладает правами на ресурс и каковы эти права. Например, для одного пользователя в ACL каталога файловой системы могут быть указаны права на чтение, а для другого пользователя - права на чтение и запись.

Основные понятия Авторизация / Проверка прав доступа (Authorization / Rights verification) — процесс установления системой соответствия запрошенных прав доступа к ресурсу и фактических прав пользователя на ресурс и формирования управляемой реакции: разрешить или отвергнуть доступ пользователя к ресурсу. Например, пользователь выполняет операцию открытия файла на запись (запрашиваемые права), обладая при этом только правом просмотра (фактические права). Система запретит выполнение операции, мотивируя свое поведение недостатком прав у пользователя.

Функции администратора ИВС Администратор ИВС обеспечивает надлежащую работоспособность ИВС, выполняя определенные функции (или должностные обязанности). К основным функциям администратора относятся:

1. Управление учетными записями пользователей. Включает регистрацию пользователей, контроль использования учетных записей, задание ограничений на использование учетных записей, блокирование временно не использующихся учетных записей, удаление учетных записей.

2. Управление доступом к ресурсам. Включает создание логических и регистрацию физических ресурсов ИВС, обеспечение управляемого доступа к ресурсам с помощью списков управления доступом, контроль использования ресурсов.

3. Обеспечение сохранности, секретности и актуальности данных. Включает создание и поддержание нужного количества резервных копий, восстановление при необходимости данных, защиту данных от

несанкционированного доступа (путем установки соответствующих прав доступа), своевременное обновление данных.

4. Установка и сопровождение программного и аппаратного обеспечения. Включает установку ОС на серверах и рабочих станциях ИВС, установку серверного и клиентского ПО, установку дополнительного аппаратного обеспечения; настройку и контроль работоспособности программного и аппаратного обеспечения.

1.2. Цели и задачи администрирования ПОИБ. Объекты администрирования

Обеспечение информационной безопасности осуществляется системой обеспечения информационной безопасности, которая состоит из технической, организационной и правовой защиты информации. Данные мероприятия проводятся руководством организации в условиях воздействия угроз непрерывно.

Администрирование информационными системами – процесс, направленный на поддержание в надлежащем состоянии и приведение в соответствие целям и задачам предприятия или организации. В целях достижения данной цели системное администрирование должно отвечать следующим критериям:

– минимальные затраты времени и ресурсов, направленные на управление системой;

– максимальная производительность и продуктивность системы.

Обеспечение информационной безопасности представляет собой деятельность по созданию препятствий и предотвращению реализации негативных воздействий (угроз) на жизненно важные интересы личности, общества и государства в информационной сфере.

В состав средств обеспечения информационной безопасности включаются:

– финансовое и материальное обеспечение информационной безопасности;

– нормативно-правовое обеспечение информационной безопасности;

– средства технологического обеспечения информационной безопасности.

Средства финансового и материального обеспечения обычно определяются федеральными целевыми программами, в рамках которых выделяются бюджетные источники денежных средств на разработку и реализацию мер противодействия угрозам информационной безопасности.

Средства технологического обеспечения информационной безопасности включают совокупность технических и программных средств защиты информации, информационно-телекоммуникационных систем и других систем обработки информации от несанкционированного доступа, а также несанкционированного выполнения операций с информацией.

В состав технологического обеспечения включается и методическое обеспечение использования технических программных средств защиты информации. Методическое обеспечение — это совокупность организационно-технических норм, которые разрабатываются в целях регулирования процессов обеспечения информационной безопасности информационной инфраструктуры. Такие нормы входят в стандарты, предназначенные для взаимодействия между производителями, потребителями и экспертами в области квалификации и сертификации современных технических средств и высоких технологий. К методическим средствам относятся и криптографические средства защиты информации.

Для успешного функционирования орган внутренних дел должен определить и осуществить управление многочисленными видами деятельности. Деятельность, использующая ресурсы и управляемая в целях преобразования входов в выходы, может быть рассмотрена как процесс. Часто выход одного процесса непосредственно формирует вход для следующего процесса.

Использование внутри органа внутренних дел системы процессов наряду с идентификацией и взаимодействием этих процессов, а также управление процессов могут быть определены как «процессный подход».

Согласно процессному подходу применительно к управлению информационной безопасностью особую значимость для органа внутренних дел имеют следующие факторы:

– понимание требований информационной безопасности органа внутренних дел и необходимости установления политики и целей информационной безопасности;

– внедрение и использование мер управления для управления рисками информационной безопасности общих рисков органа внутренних дел;

– мониторинг и проверка производительности и эффективности системы управления информационной безопасностью;

– непрерывное улучшение системы управления информационной безопасностью, основанное на результатах объективных измерений.

В ГОСТ Р ИСО/МЭК 27001-2006 [2] представлена модель «Планирование (Plan) - Осуществление (Do) - Проверка (Check) - Действие (Act)» (PDCA), которая может быть применена при структурировании всех процессов системы управления информационной безопасностью. На рисунке 1.1 показано, как система управления информационной безопасностью, используя в качестве входных данных требования информационной безопасности и ожидаемые результаты заинтересованных сторон, с помощью необходимых действий и процессов выдает выходные данные по результатам обеспечения информационной безопасности, которые соответствуют этим требованиям и ожидаемым результатам. Рисунок 1.1 иллюстрирует также связи между процессами.

Мероприятия и действия по обеспечению информационной безопасности осуществляются путем реализации процессов управления. Функцией, объединяющей процессы обеспечения, управления и менеджмента, является руководство информационной безопасности.

Разработка системы менеджмента (администрирования) информационной безопасности включает в себя разработку политики, установление целей, процессов и процедур системы, относящихся к управлению рисками и совершенствованию информационной безопасности, для достижения результатов, отвечающих требованиям общей политики и целям организации.

Внедрение и организация функционирования системы администрирования информационной безопасности проявляется во внедрении и применении политик информационной безопасности, мер менеджмента, процессов и процедур системы администрирования информационной безопасности.

На этапе проведения мониторинга и анализа системы администрирования информационной безопасности происходит оценка результативности процессов, в том числе, по возможности, количественная, касаемо требований политики, целей безопасности и практического опыта функционирования СМИБ, а также информирование руководства о результатах последующего анализа.

На этапе сопровождения (поддержки и улучшения системы администрирования информационной безопасности) проводятся корректирующие и предупредительные мероприятия, основанные на результатах внутреннего аудита и иной информации, а также анализа со стороны руководства в целях обеспечения непрерывного совершенствования системы администрирования информационной безопасности.

Современные стандарты в области управления информационной безопасностью предполагают использовать процессный подход для разработки, внедрения, обеспечения функционирования, мониторинга, анализа, поддержки и улучшения системы управления информационной безопасностью органа внутренних дел.

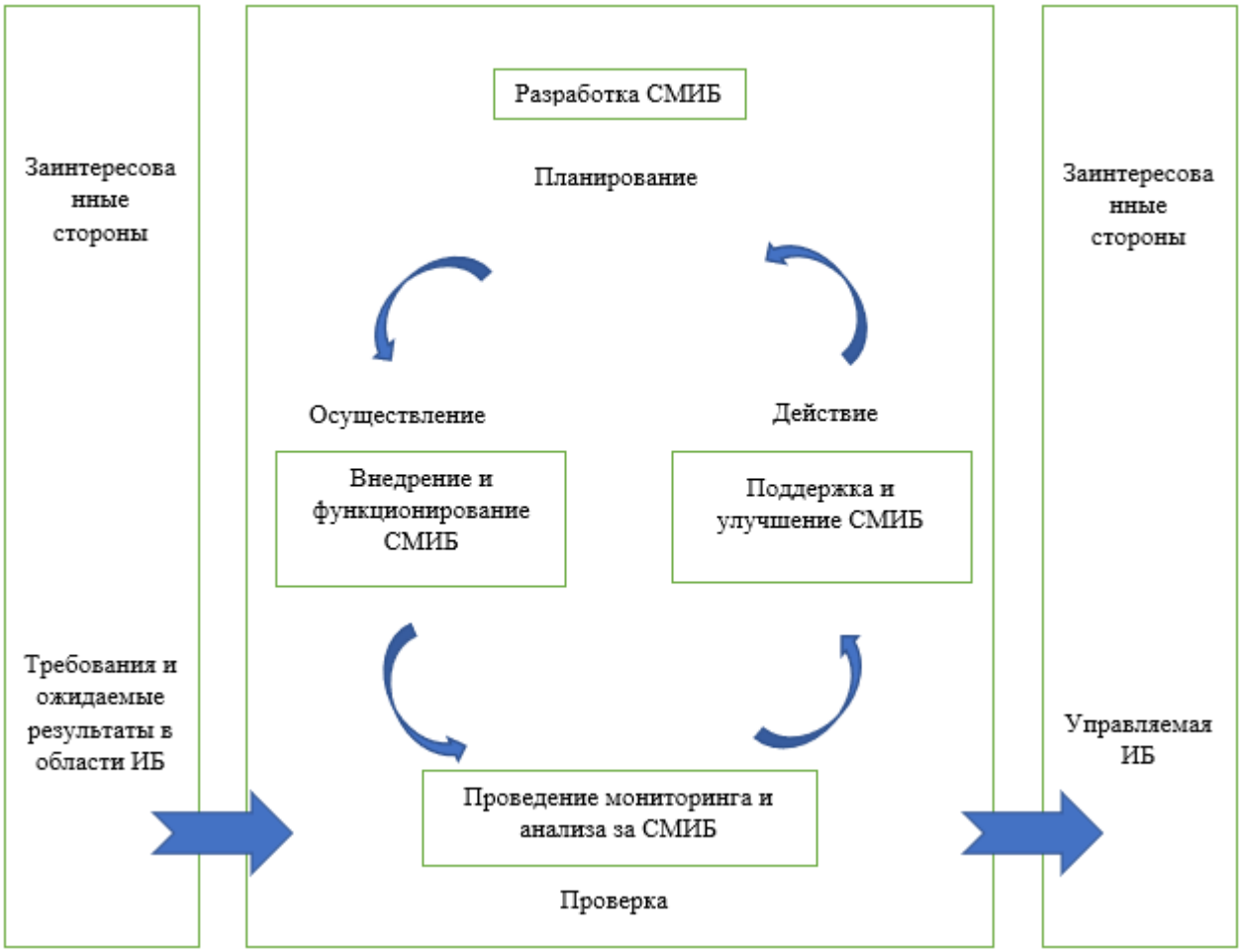

Рисунок 1.1. Модель системы управления информационной безопасностью.

1.3. Технологии обеспечения информационной безопасности в компьютерных системах и сетях

Основной особенностью любой сетевой структуры (системы) является то, что ее компоненты распределены в пространстве и связь между ними осуществляется физически при помощи сетевых соединений (коаксиальный кабель, витая пара, оптоволокно и т.п.) и программно – с помощью механизма сообщений. При этом все управляющие сообщения и данные, пересылаемые между объектами распределенной вычислительной системы, передаются по сетевым соединениям в виде пакетов обмена.

Сетевые системы характерны тем, что наряду с обычными (локальными) непреднамеренными действиями и атаками, осуществляемыми в пределах одной

компьютерной системы, к ним применим специфический вид атак, обусловленный распределенностью ресурсов и информации в пространстве. Это так называемые сетевые (удаленные) атаки (Remote Network Attacks). Они характеризуются, во-первых, тем, что злоумышленник может находиться за тысячи километров от атакуемого объекта, и, во-вторых, тем, что нападению может подвергаться не конкретный компьютер, а информация, передающаяся по сетевым соединениям. С развитием локальных и глобальных сетей именно удаленные атаки становятся лидирующими как по числу попыток, так и по успешности их применения. Соответственно обеспечение безопасности вычислительных сетей с точки зрения противостояния удаленным атакам приобретает первостепенное значение.

Современные сервисы безопасности функционируют в распределенной среде, поэтому необходимо учитывать наличие как локальных, так и сетевых угроз. В качестве общих выделяют следующие угрозы:

– обход злоумышленником защитных средств;

– осуществление злоумышленником физического доступа к вычислительной установке, на которой функционирует сервис безопасности;

– ошибки администрирования, в частности неправильная установка, ошибки при конфигурировании и т.п.;

– переход сервиса в небезопасное состояние в результате сбоя или отказа, при начальной загрузке, в процессе или после перезагрузки;

– маскарад пользователя (попытка злоумышленника выдать себя за уполномоченного пользователя, в частности, за администратора). В распределенной среде маскарад может реализовываться путем подмены исходного адреса или воспроизведения ранее перехваченных данных идентификации/аутентификации;

– маскарад сервера (попытка злоумышленника выдать свою систему за легальный сервер). Следствием маскарада сервера может стать навязывание пользователю ложной информации или получение от пользователя конфиденциальной информации;

– использование злоумышленником чужого сетевого соединения или интерактивного сеанса (например, путем доступа к оставленному без присмотра терминалу);

– несанкционированное изменение злоумышленником конфигурации сервиса и (или) конфигурационных данных;

– нарушение целостности программной конфигурации сервиса, в частности внедрение троянских компонентов, или получение контроля над сервисом;

– НСД к конфиденциальной (например, регистрационной) информации, в том числе несанкционированное расшифрование зашифрованных данных;

– несанкционированное изменение данных (например, регистрационной информации), в том числе таких, целостность которых защищена криптографическими методами;

– НСД к данным (на чтение и (или) изменение) в процессе их передачи по сети;

– анализ потоков данных с целью получения конфиденциальной информации;

– перенаправление потоков данных (в частности, на системы, контролируемые злоумышленником);

– блокирование потоков данных;

– повреждение или утрата регистрационной, конфигурационной или иной информации, влияющей на безопасность функционирования сервиса (например, из-за повреждения носителей или переполнения регистрационного журнала);

– агрессивное потребление злоумышленником ресурсов, в частности ресурсов протоколирования и аудита, а также полосы пропускания;

– сохранение остаточной информации в многократно используемых объектах.

Можно выделить следующие виды средств защиты информации от перечисленных угроз:

– средства управления обновлениями программных компонент;
– межсетевые экранированы;
– средства построения VPN;
– контроля доступа;
– средства обнаружения и предотвращения вторжений и аномалий;
– резервного копирования и архивирования;
– централизованного управления безопасностью;
– средства предотвращения вторжений на уровне серверов;
– аудита и мониторинга средств безопасности;
– контроля деятельности сотрудников в сети Интернет;
– анализа содержимого почтовых сообщений;
– анализа защищенности ИС;
– защиты от спама;
– защиты от атак класса «Отказ в обслуживании»;
– контроля целостности;
– инфраструктуры открытых ключей;
– усиленной аутентификации;
– и пр.

На основании политики информационной безопасности и указанных средств защиты информации разрабатываются конкретные процедуры защиты, включающие в себя распределение ответственности за их выполнение. Процедуры безопасности так же важны, как и политики безопасности. Если политики безопасности определяют, что должно быть защищено, то процедуры определяют, как защитить информационные ресурсы компании и кто конкретно должен разрабатывать, внедрять данные процедуры и контролировать их исполнение.

1.3.1. Диагностика сетевого подключения

Одна из важнейших подсистем, отвечающая за связь любого сервера с внешним миром — сетевая. Через сетевые интерфейсы поступают запросы от удаленных систем и через эти же интерфейсы направляются ответы, что позволяет налаживать коммуникацию и предоставлять/получать сервисы. В связи с этим особенно важно уметь производить диагностику и мониторинг сети хотя бы на базовом уровне, чтобы выявлять проблемы и вносить корректировки в конфигурацию в случае необходимости.

Диагностика сетевой связности (ping, arp, traceroute).

В данном параграфе будет описано использование протокола IP версии 4. Согласно стандартам, определяющим работу этого протокола, каждое устройство, подключенное к сети, должно иметь как минимум IP-адрес и маску подсети — параметры, которые позволяют уникально идентифицировать устройство в пределах определенной сети. В такой конфигурации устройство может обмениваться сетевыми пакетами с другими устройствами в пределах той же самой логической сети. Если к этому набору параметров добавить адрес шлюза по умолчанию — наш сервер сможет связываться с хостами, находящимися за пределами локального адресного пространства.

В случае каких-либо сетевых проблем в первую очередь проверяем, не сбились ли настройки сетевого интерфейса. Например, команды ip addr или ifconfig выведут IP-адрес и маску сети (рис. 1.2).

```
root@Ubuntu1804x64:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:01:36:6b brd ff:ff:ff:ff:ff:ff
    inet 46.243.183.90/24 brd 46.243.183.255 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe01:366b/64 scope link
       valid_lft forever preferred_lft forever
```

Рисунок 1.2. Проверка настроек сетевого интерфейса.

В выводе команды виден перечень сетевых интерфейсов, распознанных операционной системой. Интерфейс **lo** — это псевдоинтерфейс (loopback). Он не используется в реальных взаимодействиях с удаленными хостами, а вот интерфейс с именем **ens192** — то, что нам нужно (именование сетевых интерфейсов различается в разных ветках и версиях ОС Linux). IP-адрес и маска сети, назначенные этому интерфейсу, указаны в поле **inet** — /24 после адреса обозначают 24-битную маску 255.255.255.0.

Теперь проверим, указан ли шлюз по умолчанию. Команды **ip route** или **route** покажут имеющиеся маршруты (рис. 1.3).

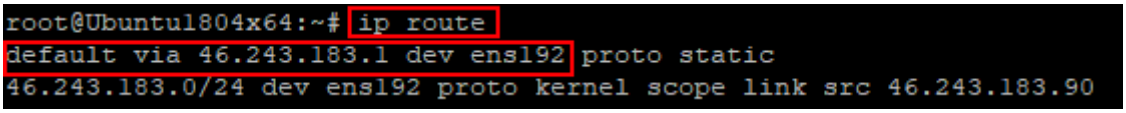

Рисунок 1.3. Проверка маршрута.

В таблице маршрутизации мы видим, что имеется маршрут по умолчанию (обозначается либо ключевым словом **default**, либо адресом 0.0.0.0). Все пакеты, предназначенные для внешних сетей, должны направляться на указанный в маршруте адрес через обозначенный сетевой интерфейс.

Если в настройках интерфейса есть ошибки, их необходимо исправить. Если же все верно — приступаем к диагностике с помощью утилиты ping. Данная команда отправляет специальные сетевые пакеты на удаленный IP-адрес (ICMP Request) и ожидает ответные пакеты (ICMP Reply). Таким образом можно проверить сетевую связность — маршрутизируются ли сетевые пакеты между IP-адресами отправителя и получателя.

Синтаксис команды **ping IP/имя опции** (рис. 1.4).

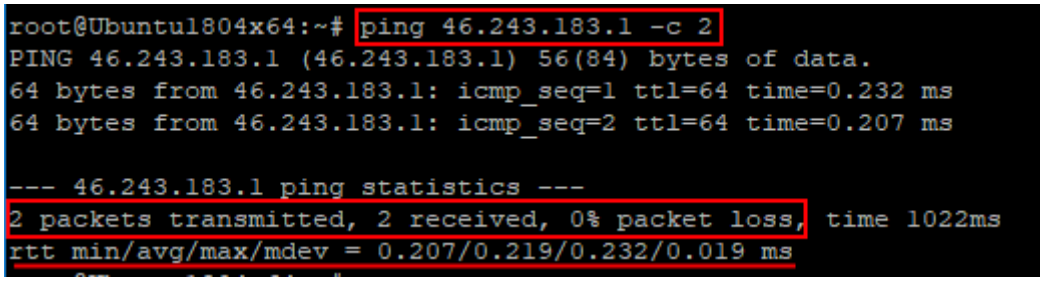

Рисунок 1.4. Синтаксис команды.

В данном случае видно, что на оба сетевых пакета, отправленных на адрес нашего шлюза по умолчанию, получены ответы, потерь нет. Это значит, что на уровне локальной сети со связностью все в порядке. Помимо количества полученных/потерянных сетевых пакетов мы можем увидеть время, которое было затрачено на прохождение запроса и ответа – параметр RTT (Round Trip Time). Этот параметр может быть очень важен при диагностике проблем, связанных с нестабильностью связи и скоростью соединения.

Часто используемые параметры:

– **ping –c количество** — указать количество пакетов, которое будет отправлено адресату (по умолчанию пакеты отправляются до тех пор, пока пользователь не прервет выполнение команды. Этот режим можно использовать, чтобы проверить стабильность сетевого соединения. Если параметр RTT будет сильно изменяться в ходе проверки, значит где-то на протяжении маршрута есть проблема);

– **ping –s количество** — указать размер пакета в байтах. По умолчанию проверка производится малыми пакетами. Чтобы проверить работу сетевых устройств с пакетами большего размера, можно использовать этот параметр;

– **ping –I интерфейс** — указать сетевой интерфейс, с которого будет отправлен запрос (актуально при наличии нескольких сетевых интерфейсов и необходимости проверить прохождение пакетов по конкретному сетевому маршруту).

В случае, если при использовании команды ping пакеты от шлюза (или другого хоста, находящегося в одной локальной сети с сервером-отправителем) в ответ не приходят, стоит проверить сетевую связность на уровне Ethernet. Здесь для коммуникации между устройствами используются так называемые MAC-адреса сетевых интерфейсов. За разрешение Ethernet-адресов отвечает протокол ARP (Address Resolution Protocol) и с помощью одноименной утилиты мы можем проверить корректность работы на этом уровне. Запустим команду **arp –n** и проверим результат (рис. 1.5).

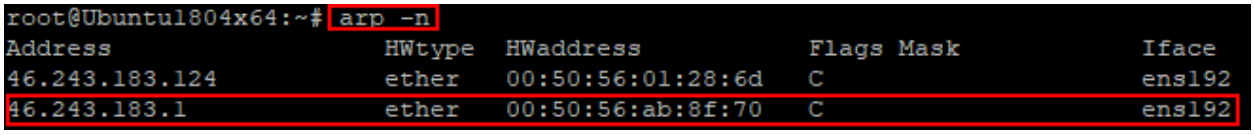

Рисунок 1.5. Команда arp –n.

Команда выведет список IP-адресов (так как был использован аргумент –n), и соответствующие им MAC-адреса хостов, находящиеся в одной сети с нашим сервером. Если в этом списке есть IP, который мы пытаемся пинговать, и соответствующий ему MAC, значит сеть работает и, возможно, ICMP-пакеты, которые использует команда ping, просто блокируются файрволом (либо со стороны отправителя, либо со стороны получателя).

Часто используемые параметры:

– **arp –n** — вывод содержимого локального arp-кэша в числовом формате. Без этой опции будет предпринята попытка определить символические имена хостов;

– **arp –d адрес** — удаление указанного адреса из кэша. Это может быть полезно для проверки корректности разрешения адреса. Чтобы убедиться, что в настоящий момент времени адрес разрешается корректно, можно удалить его из кэша и снова запустить ping. Если все работает правильно, адрес снова появится в кэше.

Если все предыдущие шаги завершены корректно, проверяем работу маршрутизатора — запускаем ping до сервера за пределами нашей сети, например, 8.8.8.8 (DNS-сервис от Google). Если все работает корректно, получаем результат (рис. 1.6).

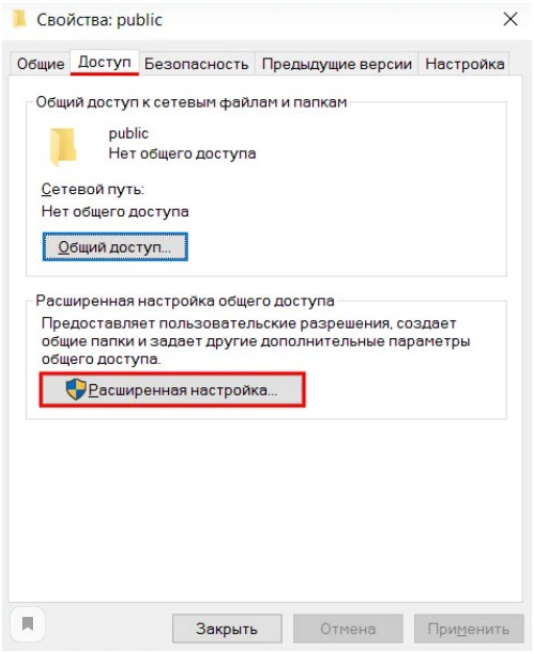

Рисунок 1.6. Проверка работы маршрутизатора.

В случае проблем на этом шаге, может помочь утилита traceroute, которая используя ту же логику запросов и ответов помогает увидеть маршрут, по которому движутся сетевые пакеты. Запускаем traceroute 8.8.8.8 –n и изучаем вывод программы (рис. 1.7).

```
root@Ubuntu1804x64:~# traceroute 8.8.8.8 -n
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  46.243.183.1  0.148 ms  0.232 ms  0.330 ms
 2  185.255.76.125  1.168 ms  1.509 ms  1.504 ms
 3  195.50.5.213  1.719 ms  1.595 ms  1.809 ms
 4  185.11.76.65  1.195 ms 185.11.76.61  1.181 ms 185.11.76.65  1.263 ms
 5  * 80.81.192.108  32.477 ms  33.795 ms
 6  108.170.251.193  38.108 ms 108.170.250.130  33.579 ms 108.170.250.66  17.752 ms
 7  216.239.48.43  36.602 ms 216.239.50.2  72.547 ms 216.239.50.132  46.453 ms
 8  8.8.8.8  39.419 ms  37.371 ms  34.905 ms
```

Рисунок 1.7. Работа утилиты traceroute.

Первым маршрутизатором на пути пакета должен быть локальный шлюз по умолчанию. Если дальше него пакет не уходит, возможно проблема в конфигурации маршрутизатора и нужно разбираться с ним. Если пакеты теряются на дальнейших шагах, возможно, есть проблема в промежуточной сети.

А, возможно, промежуточные маршрутизаторы не отсылают ответные пакеты. В этом случае можно переключиться на использование другого протокола в traceroute.

Часто используемые опции:

– **traceroute –n** — вывод результата в числовом формате вместо символических имен промежуточных узлов;

– **traceroute –I** — использование ICMP-протокола при отслеживании маршрута. По умолчанию используются UDP-датаграммы;

– **traceroute –s** адрес— указать адрес источника для исходящего сетевого пакета;

– **traceroute –i** интерфейс— указать сетевой интерфейс, с которого будут отправляться пакеты.

Диагностика разрешения имен (nslookup, dig).

Разобравшись с сетевой связностью и маршрутизацией, приходим к следующему этапу — разрешение доменных имен. В большинстве случаев в работе с удаленными сервисами мы не используем IP-адреса, а указываем доменные имена удаленных ресурсов. За перевод символических имен в IP-адреса отвечает служба DNS — это сеть серверов, которые содержат актуальную информацию о соответствии имен и IP в пределах доверенных им доменных зон.

Самый простой способ проверить работает ли разрешение имен — запустить утилиту ping с указанием доменного имени вместо IP-адреса (например, ping ya.ru). Если ответные пакеты от удаленного сервера приходят, значит все работает как надо. В противном случае нужно проверить прописан ли DNS-сервер в сетевых настройках и удается ли получить от него ответ.

Способы выяснения какой DNS-сервер использует наш сервер различаются в зависимости от используемой версии и дистрибутива ОС Linux. Например, если ОС используется Network Manager для управления сетевыми интерфейсами (CentOS, RedHat и др.), может помочь вывод команды nmcli.

В настройках сетевого интерфейса, в разделе DNS configuration, видно IP-адрес сервера. В Ubuntu 18.04 и выше, использующих Netplan, используем команду **systemd-resolve —status**.

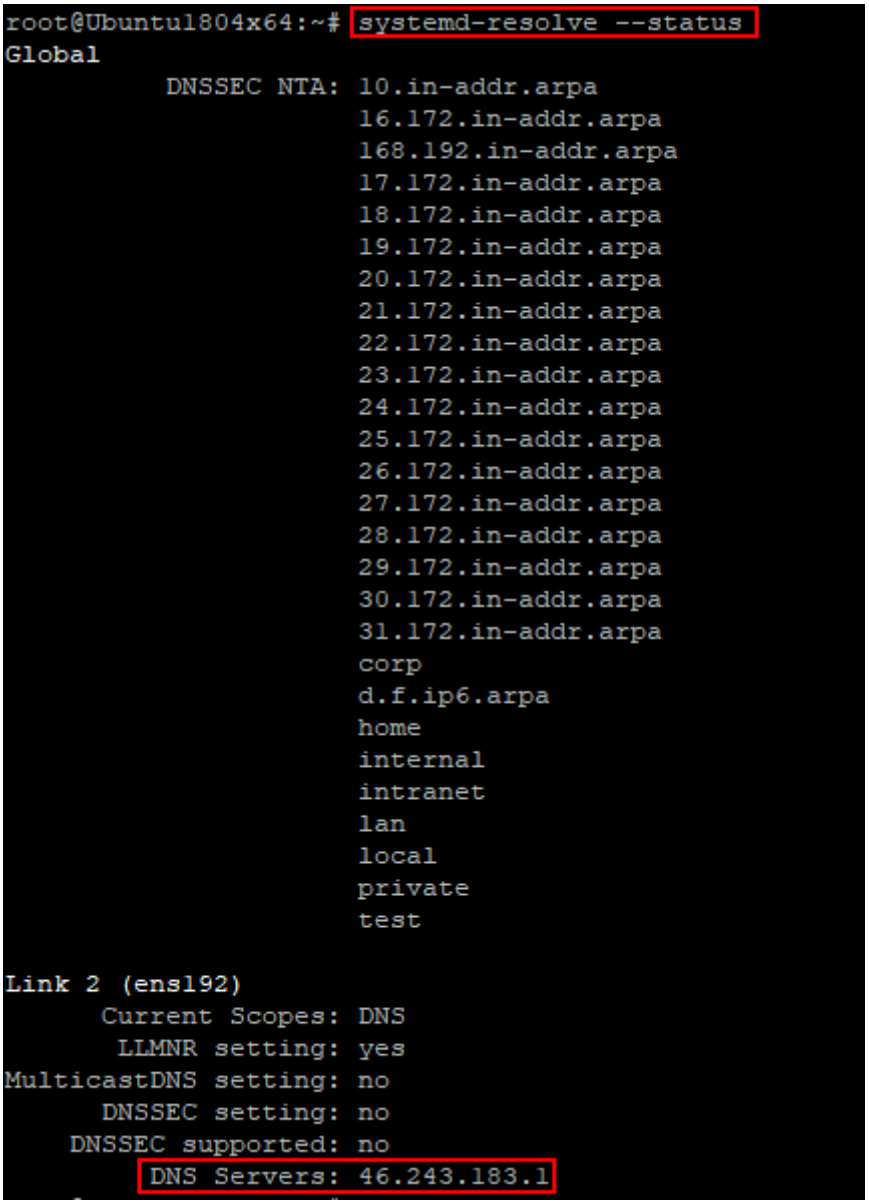

Рисунок 1.8. Команда systemd-resolve —status.

Используемый сервер также будет указан в настройках интерфейса, в разделе DNS Servers. В более старых версиях Ubuntu потребуется проверить содержимое файлов **/etc/resolve.conf** и **/etc/network/interfaces**.

Проверить работу сервиса разрешения имен нам помогут утилиты nslookup или dig. Функционально они почти идентичны: G-вывод утилиты dig содержит

больше диагностической информации и гибко регулируется, но это далеко не всегда нужно. Поэтому используйте ту утилиту, которая удобна в конкретной ситуации. Если эти команды недоступны, потребуется доставить пакеты

для CentOS/RedHat:

yum install bind-utils

для Debian/Ubuntu:

sudo apt install dnsutils

После успешной установки сделаем тестовые запросы:

dig ya.ru

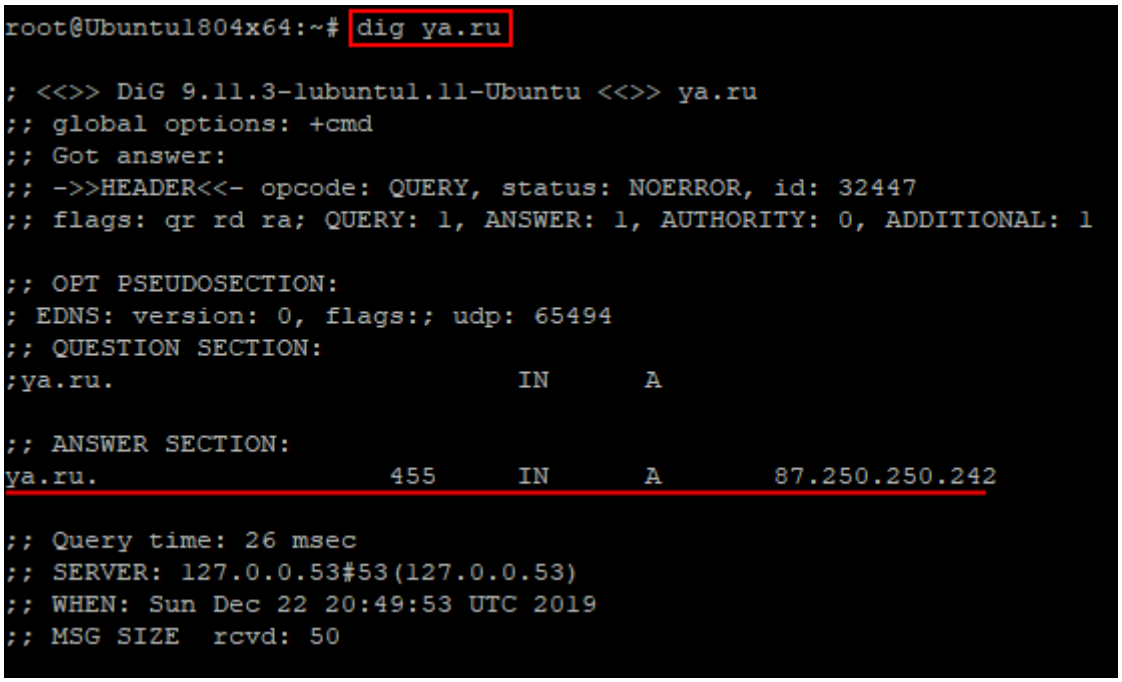

Рисунок 1.9. Команда systemd-resolve —status.

В разделе **Answer Section** видим ответ от DNS сервера — IP-адрес для A-записи с доменным именем ya.ru. Разрешение имени работает корректно:

nslookup ya.ru

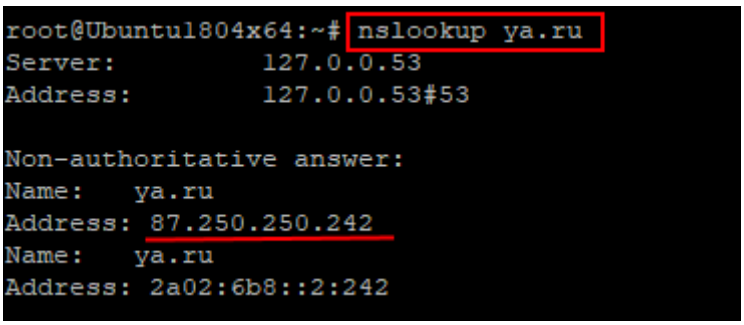

Рисунок 1.10. Подтверждение корректной работы.

Аналогичный запрос утилитой **nslookup** выдает более компактный вывод, но вся нужная сейчас информация в нем присутствует.

Что же делать, если в ответе отсутствует IP-адрес? Возможно, DNS-сервер недоступен. Для проверки можно отправить тестовый запрос на другой DNS-сервер. Обе утилиты позволяют эти сделать. Направим тестовый запрос на DNS-сервер Google:

dig @8.8.8.8 ya.ru

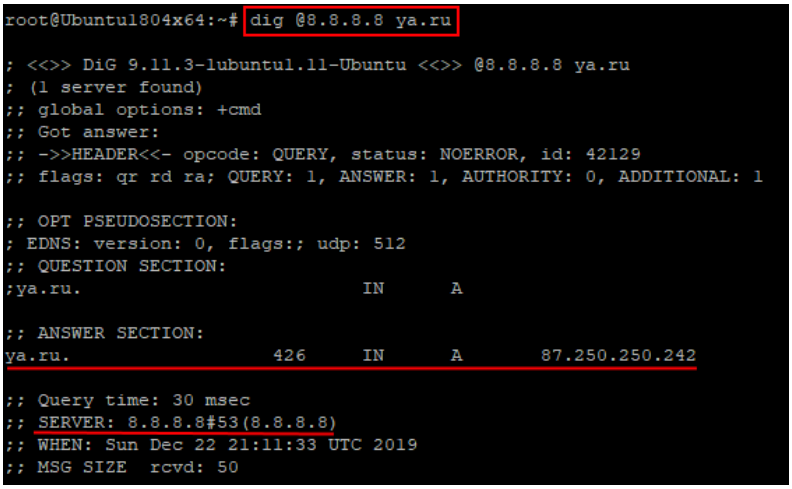

Рисунок 1.11. Отправка тестового запроса 1.

nslookup ya.ru 8.8.8.8

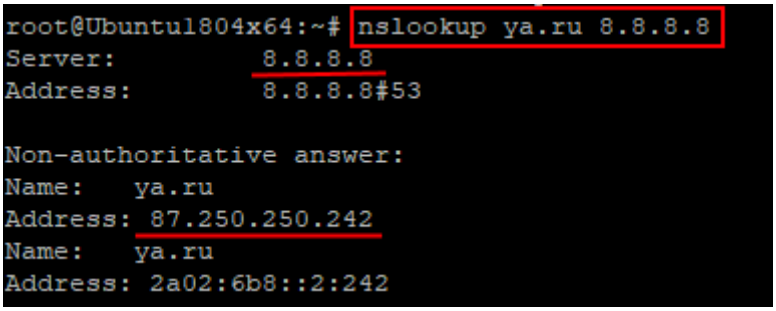

Рисунок 1.12. Отправка тестового запроса 2.

Если имена разрешаются публичным DNS-сервером корректно, а установленным по умолчанию в ОС нет, вероятно, есть проблема в работе этого DNS-сервера. Временным решением данной проблемы может быть использование публичного DNS-сервера в качестве сервера для разрешения имен в операционной системе. В том случае, если разрешение имен не работает ни через локальный, ни через публичный DNS сервер — стоит проверить не блокируют ли правила файрвола отправку на удаленный порт 53 TCP/UDP пакетов (именно на этом порту DNS-серверы принимают запросы).

Часто используемые параметры:

nslookup имя сервер — разрешить доменное имя, используя альтернативый сервер;

nslookup –type=тип имя — получить запись указанного типа для доменного имени (например, **nslookup -type=mx ya.ru** – получить MX-записи для домена ya.ru);

dig @сервер имя — разрешить доменное имя, используя альтернативый сервер;

dig имя тип — получить запись указанного типа для доменного имени (например, **dig ya.ru mx** — получить MX-записи для домена ya.ru).

2. АДМИНИСТРИРОВАНИЕ ЭЛЕМЕНТОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СЕТЕЙ

2.1. Задачи и цели сетевого администрирования

2.1.1. Общие сведения и основа функционирования сетевых ресурсов

Современные корпоративные информационные системы по своей природе всегда являются распределенными системами. Рабочие станции пользователей, серверы приложений, серверы баз данных и прочие сетевые узлы распределены по большой территории. В крупной компании офисы и площадки соединены различными видами коммуникаций, использующих различные технологии и сетевые устройства. Главная задача сетевого администратора — обеспечить надежную, бесперебойную, производительную и безопасную работу всей этой сложной системы.

Стоит рассматривать рассматривать сеть как совокупность программных, аппаратных и коммуникационных средств, обеспечивающих эффективное распределение вычислительных ресурсов. Все сети можно условно разделить на 3 категории:

- локальные сети (LAN, Local Area Network);
- глобальные сети (WAN, Wide Area Network);
- городские сети (MAN, Metropolitan Area Network).

Глобальные сети позволяют организовать взаимодействие между абонентами на больших расстояниях. Эти сети работают на относительно низких скоростях и могут вносить значительные задержки в передачу информации. Протяженность глобальных сетей может составлять тысячи километров. Поэтому они так или иначе интегрированы с сетями масштаба страны.

Городские сети позволяют взаимодействовать на территориальных образованиях меньших размеров и работают на скоростях от средних до высоких. Они меньше замедляют передачу данных, чем глобальные, но не могут

обеспечить высокоскоростное взаимодействие на больших расстояниях. Протяженность городских сетей находится в пределах от нескольких километров до десятков и сотен километров.

Локальные сети обеспечивают наивысшую скорость обмена информацией между компьютерами. Типичная локальная сеть занимает пространство в одно здание. Протяженность локальных сетей составляет около одного километра. Их основное назначение состоит в объединении пользователей (как правило, одной компании или организации) для совместной работы.

Механизмы передачи данных в локальных и глобальных сетях существенно отличаются. Глобальные сети ориентированы на соединение — до начала передачи данных между абонентами устанавливается соединение (сеанс). В локальных сетях используются методы, не требующие предварительной установки соединения, — пакет с данными посылается без подтверждения готовности получателя к обмену.

Кроме разницы в скорости передачи данных, между этими категориями сетей существуют и другие отличия. В локальных сетях каждый компьютер имеет сетевой адаптер, который соединяет его со средой передачи. Городские сети содержат активные коммутирующие устройства, а глобальные сети обычно состоят из групп мощных маршрутизаторов пакетов, объединенных каналами связи. Кроме того, сети могут быть частными или сетями общего пользования.

Сетевая инфраструктура строится из различных компонентов, которые условно можно разнести по следующим уровням:

- кабельная система и средства коммуникаций;
- активное сетевое оборудование;
- сетевые протоколы;
- сетевые службы;
- сетевые приложения.

Каждый из этих уровней может состоять из различных подуровней и компонентов. Например, кабельные системы могут быть построены на основе коаксиального кабеля ("толстого" или тонкого"), витой пары (экранированной и

неэкранированной), оптоволокна. Активное сетевое оборудование включает в себя такие виды устройств, как повторители (репитеры), мосты, концентраторы, коммутаторы, маршрутизаторы. В корпоративной сети может быть использован богатый набор сетевых протоколов: TCP/IP, SPX/IPX, NetBEUI, AppleTalk и др.

Основу работы сети составляют так называемые сетевые службы (или сервисы). Базовый набор сетевых служб любой корпоративной сети состоит из следующих служб:

- службы сетевой инфраструктуры DNS, DHCP, WINS;
- службы файлов и печати;
- службы каталогов (например, Novell NDS, MS Active Directory);
- службы обмена сообщениями;
- службы доступа к базам данных.

Самый верхний уровень функционирования сети — сетевые приложения.

Сеть позволяет легко взаимодействовать друг с другом самым различным видам компьютерных систем благодаря стандартизованным методам передачи данных, которые позволяют скрыть от пользователя все многообразие сетей и машин.

Все устройства, работающие в одной сети, должны общаться на одном языке – передавать данные в соответствии с общеизвестным алгоритмом в формате, который будет понят другими устройствами. Стандарты – ключевой фактор при объединении сетей.

Для более строгого описания работы сети разработаны специальные модели. В настоящее время общепринятыми моделями являются модель OSI (Open System Interconnection) и модель TCP/IP (или модель DARPA). Обе модели будут рассмотрены в данном разделе ниже.

Прежде чем определить задачи сетевого администрирования в сложной распределенной корпоративной сети, сформулируем определение термина "корпоративная сеть" (КС). Слово "корпорация" означает объединение предприятий, работающих под централизованным управлением и решающих общие задачи. Корпорация является сложной, многопрофильной структурой и

вследствие этого имеет распределенную иерархическую систему управления. Кроме того, предприятия, отделения и административные офисы, входящие в корпорацию, как правило, расположены на достаточном удалении друг от друга. Для централизованного управления таким объединением предприятий используется корпоративная сеть.

Основная задача КС заключается в обеспечении передачи информации между различными приложениями, используемыми в организации. Под приложением понимается программное обеспечение, которое непосредственно нужно пользователю, например, бухгалтерская программа, программа обработки текстов, электронная почта и т.д. Корпоративная сеть позволяет взаимодействовать приложениям, зачастую расположенным в географически различных областях, и обеспечивает доступ к ним удаленных пользователей. На рисунке 2.1 показана обобщенная функциональная схема корпоративной сети.

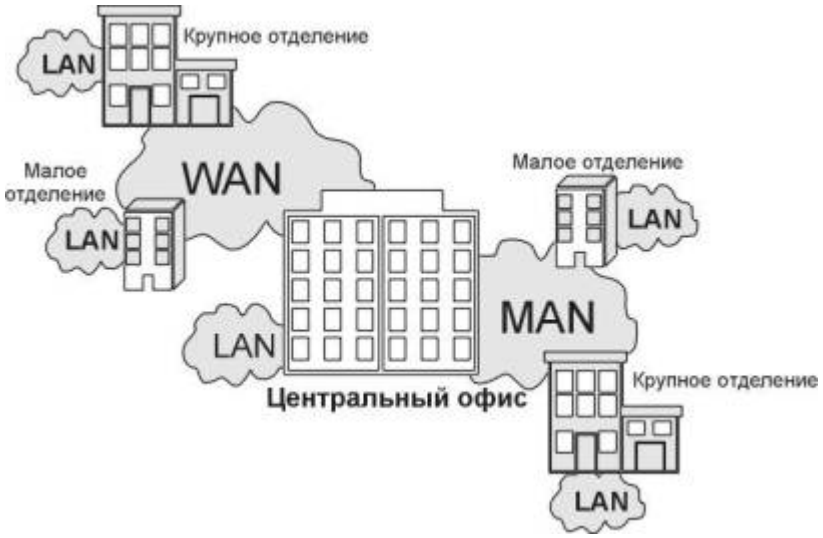

Рисунок 2.1. Обобщенная схема КС.

Обязательным компонентом корпоративной сети являются локальные сети, связанные между собой.

2.1.2. Задачи сетевого администрирования в сложной распределенной сети

Планирование сети. Несмотря на то, что планированием и монтажом больших сетей обычно занимаются специализированные компании-интеграторы, сетевому администратору часто приходится планировать

определенные изменения в структуре сети — добавление новых рабочих мест, добавление или удаление сетевых протоколов, добавление или удаление сетевых служб, установка серверов, разбиение сети на сегменты и т.д. Данные работы должны быть тщательно спланированы, чтобы новые устройства, узлы или протоколы включались в сеть или исключались из нее без нарушения целостности сети, без снижения производительности, без нарушения инфраструктуры сетевых протоколов, служб и приложений.

Установка и настройка сетевых узлов (устройств активного сетевого оборудования, персональных компьютеров, серверов, средств коммуникаций). Данные работы могут включать в себя — замену сетевого адаптера в ПК с соответствующими настройками компьютера, перенос сетевого узла (ПК, сервера, активного оборудования) в другую подсеть с соответствующим изменениями сетевых параметров узла, добавление или замена сетевого принтера с соответствующей настройкой рабочих мест.

Установка и настройка сетевых протоколов. Данная задача включает в себя выполнение таких работ — планирование и настройка базовых сетевых протоколов корпоративной сети, тестирование работы сетевых протоколов, определение оптимальных конфигураций протоколов.

Установка и настройка сетевых служб. Корпоративная сеть может содержать большой набор сетевых служб. Кратко перечислим основные задачи администрирования сетевых служб:

– установка и настройка служб сетевой инфраструктуры (службы DNS, DHCP, WINS, службы маршрутизации, удаленного доступа и виртуальных частных сетей);

– установка и настройка служб файлов и печати, которые в настоящее время составляют значительную часть всех сетевых служб;

– администрирование служб каталогов (Novell NDS, Microsoft Active Directory), составляющих основу корпоративной системы безопасности и управления доступом к сетевым ресурсам;

– администрирование служб обмена сообщениями (системы электронной почты);

– администрирование служб доступа к базам данных.

Поиск неисправностей. Сетевой администратор должен уметь обнаруживать широкий спектр неисправностей — от неисправного сетевого адаптера на рабочей станции пользователя до сбоев отдельных портов коммутаторов и маршрутизаторов, а также неправильные настройки сетевых протоколов и служб.

Поиск узких мест сети и повышения эффективности работы сети. В задачу сетевого администрирования входит анализ работы сети и определение наиболее узких мест, требующих либо замены сетевого оборудования, либо модернизации рабочих мест, либо изменения конфигурации отдельных сегментов сети.

Мониторинг сетевых узлов. Мониторинг сетевых узлов включает в себя наблюдение за функционированием сетевых узлов и корректностью выполнения возложенных на данные узлы функций.

Мониторинг сетевого трафика. Мониторинг сетевого трафика позволяет обнаружить и ликвидировать различные виды проблем: высокую загруженность отдельных сетевых сегментов, чрезмерную загруженность отдельных сетевых устройств, сбои в работе сетевых адаптеров или портов сетевых устройств, нежелательную активность или атаки злоумышленников (распространение вирусов, атаки хакеров и др.).

Обеспечение защиты данных. Защита данных включает в себя большой набор различных задач: резервное копирование и восстановление данных, разработка и осуществление политик безопасности учетных записей пользователей и сетевых служб (требования к сложности паролей, частота смены паролей), построение защищенных коммуникаций (применение протокола IPSec, построение виртуальных частных сетей, защита беспроводных сетей), планирование, внедрение и обслуживание инфраструктуры открытых ключей (PKI).

2.2. Модели межсетевого взаимодействия

2.2.1. Модель OSI

Модели межсетевого взаимодействия предназначены для формального и в то же время наглядного описания взаимодействия сетевых узлов между собой. В настоящее время наибольшее распространение получили и являются стандартами для описания межсетевого взаимодействия две сетевые модели: модель OSI и модель TCP/IP. Обе модели разбивают процесс взаимодействия сетевых узлов на несколько уровней, каждый конкретный уровень одного узла обменивается информацией с соответствующим уровнем другого узла.

Каждую из этих моделей можно представлять как объединение двух моделей:

– горизонтальная модель (на базе протоколов, обеспечивающая обмен данными одного типа между программами и процессами, работающими на одном и том же уровне на различных сетевых узлах);

– вертикальная модель (на основе услуг, предоставляемых соседними уровнями друг другу на одном сетевом узле).

В горизонтальной модели двум программам, работающими на различных сетевых узлах, требуется общий протокол для обмена данными. В вертикальной – соседние уровни обмениваются данными, выполняя необходимые преобразования с использованием соответствующих программных интерфейсов.

В 1983 году с целью упорядочения описания принципов взаимодействия устройств в сетях Международная организация по стандартизации (International Organization for Standardization, ISO) предложила семиуровневую эталонную коммуникационную модель "Взаимодействие Открытых Систем", модель OSI (Open System Interconnection).

Эталонная модель OSI сводит передачу информации в сети к семи относительно простым подзадачам.

Модель OSI стала основой для разработки стандартов на взаимодействие систем. Она определяет только схему выполнения необходимых задач, но не дает

конкретного описания их выполнения. Это описывается конкретными протоколами или правилами, разработанными для определенной технологии с учетом модели OSI. Уровни OSI могут реализовываться как аппаратно, так и программно.

Основная идея модели OSI в том, что одни и те же уровни на разных системах, не имея возможности связываться непосредственно, должны работать абсолютно одинаково. Одинаковым должен быть и сервис между соответствующими уровнями различных систем. Нарушение этого принципа может привести к тому, что информация, посланная от одной системы к другой, после всех преобразований не будет идентична исходной.

Существует семь основных уровней модели OSI (таблица 2.1). Они начинаются с физического уровня и заканчиваются прикладным. Каждый уровень предоставляет услуги для более высокого уровня. Седьмой уровень обслуживает непосредственно пользователей.

Таблица 2.1. Семиуровневая модель OSI

Семиуровневая модель OSI	
Уровень	
7. Прикладной	7. Application
6. Представления	6. Presentation
5. Сеансовый	5. Session
4. Транспортный	4. Transport
3. Сетевой	3. Network
2. Канальный	2. Data Link
1. Физический	1. Physical

Модель OSI описывает путь информации через сетевую среду от одной прикладной программы на одном компьютере до другой программы на другом компьютере. При этом пересылаемая информация проходит вниз через все уровни системы.

Уровни на разных системах не могут общаться между собой напрямую. Это умеет только физический уровень.

По мере прохождения информации вниз внутри системы она преобразуется в вид, удобный для передачи по физическим каналам связи.

Для указания адресата к этой преобразованной информации добавляется заголовок с адресом. После получения адресатом этой информации, она проходит через все уровни наверх. По мере прохождения информация преобразуется в первоначальный вид.

Каждый уровень системы должен полагаться на услуги, предоставляемые ему смежными уровнями.

1. *Физический уровень*. На данном уровне выполняется передача битов по физическим каналам (коаксиальный кабель, витая пара, оптоволокно).

2. *Канальный уровень*. Данный уровень определяет методы доступа к среде передачи данных и обеспечивает передачу кадра данных между любыми узлами в сетях с типовой топологией по физическому адресу сетевого устройства. Адреса, используемые на канальном уровне в локальных сетях, часто называют MAC-адресами (MAC — media access control, управление доступом к среде передачи данных).

3. *Сетевой уровень*. Обеспечивает доставку данных между любыми двумя узлами в сети с произвольной топологией, при этом не гарантируется надежная доставка данных от узла-отправителя к узлу-получателю. На этом уровне выполняются такие функции как маршрутизация логических адресов сетевых узлов, создание и ведение таблиц маршрутизации, фрагментация и сборка данных.

4. *Транспортный уровень*. Обеспечивает передачу данных между любыми узлами сети с требуемым уровнем надежности. Для выполнения этой задачи на транспортном уровне имеются механизмы установления соединения между сетевыми узлами, нумерации, буферизации и упорядочивания пакетов, передаваемых между узлами сети.

5. *Сеансовый уровень*. Реализует средства управления сессией, диалогом, а также предоставляет средства синхронизации в рамках процедуры обмена сообщениями, контроля над ошибками, обработки транзакций, поддержки вызова удаленных процедур RPC.

6. *Уровень представления*. На этом уровне могут выполняться различные виды преобразования данных, такие как компрессия и декомпрессия, шифровка и дешифровка данных.

7. *Прикладной уровень*. Набор сетевых сервисов, предоставляемых конечным пользователям и приложениям. Примеры таких сервисов — обмен сообщениями электронной почты, передача файлов между узлами сети, приложения управления сетевыми узлами.

Функционирование первых трех уровней, физического, канального и сетевого, обеспечивается, в основном, активным сетевым оборудованием и, как правило, реализуются следующими компонентами: сетевыми адаптерами, репитерами, мостами, концентраторами, коммутаторами, маршрутизаторами.

2.2.2. Модель TCP/IP

Модель TCP/IP называют также моделью DARPA (сокращение от Defense Advanced Research Projects Agency, организация, в которой в свое время разрабатывались сетевые проекты, в том числе протокол TCP/IP, и которая стояла у истоков сети Интернет) или моделью Министерства обороны США (модель DoD, Department of Defense, проект DARPA работал по заказу этого ведомства).

Впервые о TCP/IP было сказано в 1973 году на заседании International Network Working Group, прошедшем в Великобритании. Здесь Роберт Кан и Винт Серф выступили с проектом статьи, которая позже, в мае 1974 года, была опубликована в одном из самых престижных журналов Transactions on Communications. В статье были изложены основы будущего протокола TCP/IP.

Главная идея, предложенная авторами, состояла в том, чтобы перенести обеспечение надежности коммуникаций из сети в подключенные к ней серверы.

Идея оказалась блестящей, она пришлась по вкусу и либерально настроенным ученым, и военным одновременно. После этого протокол начал жить своей жизнью, пока еще под названием TCP. К совершенствованию нового протокола приложили руку многие инженеры и ученые, и к октябрю 1977 года его работу удалось продемонстрировать не только в ARPAnet, но и в пакетной радиосети и спутниковой сети SATNET.

Чуть позже инженеры пришли к выводу о необходимости разделить протокол на две части: так появились "близнецы-братья" TCP и IP. Часть TCP отвечает за разбиение сообщения на дейтаграммы на стороне отправителя, за сборку их на стороне получателя, обнаружение ошибок и восстановление порядка пакетов, если он был нарушен в процессе передачи. IP, или Internet Protocol, отвечает за маршрутизацию отдельных дейтаграмм.

История создания TCP/IP ведет свое начало с момента, когда министерство обороны США столкнулось с проблемой объединения большого числа компьютеров с различными ОС. В 1970 г. был разработан необходимый набор стандартов. Протоколы, разработанные на базе этих стандартов, получили обобщенное название TCP/IP.

Модель TCP/IP разрабатывалась для описания стека протоколов TCP/IP (Transmission Control Protocol/Internet Protocol). Она была разработана значительно раньше, чем модель OSI.

Формальные правила, определяющие последовательность и формат сообщений на одном уровне, называются протоколами. Иерархически организованная совокупность протоколов называется стеком коммуникационных протоколов.

Модель состоит из четырех уровней, представленных в таблице 2.2.

Таблица 2.2. Соответствие уровней протоколам модели TCP/IP

Уровень модели TCP/IP	**Протоколы**
Прикладной уровень (Application)	WWW, FTP, TFTP, SNMP, Telnet, SMTP, DNS, DHCP, WINS
Транспортный уровень (Transport)	TCP, UDP
Уровень межсетевого взаимодействия (Internet)	ARP, IP, ICMP, RIP, OSPF
Уровень сетевого интерфейса (Network Interface)	Не регламентируется спецификациями стека TCP/IP (Ethernet, Token Ring, FDDI, ATM, X.25, Frame Relay, SLIP, PPP)

Таблица 2.3. Соответствие уровней моделей OSI и TCP/IP

Уровень модели OSI	**Уровень модели TCP/IP**
7. Прикладной (Application)	1. Прикладной уровень (Application)
6. Представления (Presentation)	
5. Сеансовый (Session)	
4. Транспортный (Transport)	2. Транспортный уровень (Transport)
3. Сетевой (Network)	3. Уровень межсетевого взаимодействия (Internet)
2. Канальный (Data Link)	4. Уровень сетевого интерфейса (Network Interface)
1. Физический (Physical)	

2.2.3. Преимущества стека протоколов TCP/IP

– Основное достоинство стека протоколов TCP/IP в том, что он обеспечивает надежную связь между сетевым оборудованием от различных производителей.

– Независимость от сетевой технологии — стек только определяет элемент передачи, дейтаграмму, и описывает способ ее движения по сети.

– Всеобщая связанность — стек позволяет любой паре компьютеров, которые его поддерживают, взаимодействовать друг с другом. Каждому

компьютеру назначается логический адрес, а каждая передаваемая дейтаграмма содержит логические адреса отправителя и получателя. Промежуточные маршрутизаторы используют адрес получателя для принятия решения о маршрутизации.

– Подтверждения. Протоколы стека обеспечивают подтверждения правильности прохождения информации при обмене между отправителем и получателем.

– Стандартные прикладные протоколы. Протоколы стека TCP/IP включают в свой состав средства поддержки основных приложений, таких как электронная почта, передача файлов, удаленный доступ и т.д.

Уровни модели взаимодействия протоколов TCP/IP:

1. *Уровень сетевого интерфейса* не регламентирован спецификациями стека TCP/IP и фактически к стеку TCP/IP относят уровни с 1-го по 3-й модели TCP/IP. Данный уровень соответствует физическому и канальному уровням модели OSI.

2. *Уровень межсетевого взаимодействия*. На данном уровне функционирует целое семейство протоколов. Основная задача данного уровня — доставка пакетов от одного узла-отправителя к узлу-получателю:

– эту задачу выполняет протокол IP (Internet Protocol, протокол межсетевого взаимодействия). Протокол IP — базовый протокол стека TCP/IP и основной протокол сетевого уровня. Отвечает за передачу информации по сети. В его основе заложен дейтаграммный метод, который не гарантирует доставку пакета;

– протокол ARP (Address Resolution Protocol, протокол разрешения физических адресов) — служит связующим звеном между уровнем межсетевого взаимодействия и уровнем сетевого интерфейса. Он преобразует IP-адреса сетевых узлов в физические MAC-адреса соответствующих сетевых адаптеров. Протокол ARP предполагает, что каждое устройство знает как свой IP-адрес, так и свой физический адрес. ARP динамически связывает их и заносит в

специальную таблицу, где хранятся пары "IP-адрес – физический адрес" (обычно каждая запись в ARP-таблице имеет время жизни 10 мин.);

– протокол ICMP (Internet Control Message Protocol, протокол межсетевых управляющих сообщений) — служит для обмена информацией об ошибках. С помощью специальных пакетов ICMP сообщает сетевым узлам информацию о невозможности доставки пакета, о превышении времени жизни пакета и др.;

– протоколы RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First) служат для построения таблиц маршрутизации и вычисления маршрутов при отправке пакетов между различными IP-сетями.

3. *Транспортный уровень*:

– протокол TCP (Transmission Control Protocol, протокол управления передачей) обеспечивает, базируясь на услугах протокола IP, надежную передачу сообщений между сетевыми узлами с помощью образования соединений (сеансов) между данными узлами. Такие протоколы прикладного уровня, как HTTP и FTP, передают протоколу TCP свои данные для транспортировки. Поэтому скоростные характеристики TCP оказывают непосредственное влияние на производительность приложений. Кроме того, протокол TCP используется для обработки запросов на вход в сеть, разделения ресурсов и т.д. На протокол TCP, в частности, возложена задача управления потоками и перегрузками. Он отвечает за согласование скорости передачи данных с техническими возможностями рабочей станции-получателя и промежуточных устройств в сети;

– протокол UDP (User Datagram Protocol, протокол дейтаграмм пользователя) обеспечивает передачу прикладных пакетов дейтаграммным способом (т.е. не гарантирующим доставку пакетов). Работа этого протокола аналогична IP, но основной его задачей является связь сетевого протокола и различных приложений.

4. *Прикладной уровень*. Приложения, перечисленные в таблице 2.2, специально разрабатывались для функционирования в сетях TCP/IP:

– протоколы для формирования сетевой инфраструктуры (DNS, DHCP, WINS) будут рассмотрены в следующих разделах данного курса;

– приложения WWW (World Wide Web, Всемирная паутина) — основа для работы сегодняшней сети Интернет. Протокол FTP (File Transfer Protocol, протокол передачи файлов) реализует удаленную передачу файлов между узлами сети;

– протокол TFTP (Trivial File Transfer Protocol, простейший протокол пересылки файлов) — более простой передачи файлов, в отличие от FTP не требующий аутентификации пользователя на удаленном узле и использующий протокол UDP для передачи информации;

– протокол SNMP (Simple Network Management Protocol, простой протокол управления сетью) используется для организации управления сетевыми узлами.

2.3. Доменная система имен DNS

2.3.1. Понятие и назначение доменной системы имен (DNS)

Каждый компьютер или иное устройство, подключенное к сети Интернет (в дальнейшем хост), имеют уникальный IP-адрес. Именно по IP-адресу происходит поиск и взаимодействие устройств в сети. IP-адрес представляет собой последовательность из четырех чисел, разделенных точками. Запоминать его в таком виде довольно сложно. Для удобства запоминания и восприятия была создана доменная система имен (Domain Name System - DNS), позволяющая поставить в соответствие IP-адресу символьное имя. Символьное имя включает в себя доменное имя и имя хоста и состоит из нескольких полей, разделенных точками. Крайнее правое поле является именем домена верхнего уровня, далее, справа налево, следуют имена доменов (поддомены) более низкого уровня. Крайнее левое поле, как правило, является именем хоста.

Например:

upc.stcompany.ru - символьное имя хоста upc в домене с именем stcompany

stcompany.ru - полное доменное имя домена второго уровня

ru - домен верхнего уровня

DNS (Domain Name System) — это распределенная база данных, поддерживающая иерархическую систему имен для идентификации узлов в сети Internet. Служба DNS предназначена для автоматического поиска IP-адреса по известному символьному имени узла.

Протокол DNS является служебным протоколом прикладного уровня. Этот протокол несимметричен - в нем определены DNS-серверы и DNS-клиенты. DNS-серверы — программы, которые при обращении к ним выискивают нужный IP-адрес по введенному URL. DNS-серверы хранят часть распределенной базы данных о соответствии символьных имен и IP-адресов. Эта база данных распределена по административным доменам сети Internet. Клиенты сервера DNS знают IP-адрес сервера DNS своего административного домена и по протоколу IP передают запрос, в котором сообщают известное символьное имя и просят вернуть соответствующий ему IP-адрес. Если данные о запрошенном соответствии хранятся в базе данного DNS-сервера, то он сразу посылает ответ клиенту, если же нет - то он посылает запрос DNS-серверу другого домена, который может сам обработать запрос, либо передать его другому DNS-серверу. Все DNS-серверы соединены иерархически, в соответствии с иерархией доменов сети Internet. Клиент опрашивает эти серверы имен, пока не найдет нужные отображения. Этот процесс ускоряется из-за того, что серверы имен постоянно кэшируют информацию, предоставляемую по запросам. Клиентские компьютеры могут использовать в своей работе IP-адреса нескольких DNS-серверов, для повышения надежности своей работы.

База данных DNS имеет структуру дерева, называемого доменным пространством имен, в котором каждый домен (узел дерева) имеет имя и может содержать поддомены.. Полное доменное имя заканчивается точкой ".", обозначающей корень доменного дерева, но часто эта завершающая точка опускается. Далее (справа налево) по иерархии идут домены верхнего уровня, за ними домены второго уровня, далее третьего и т.д. Друг от друга они отделяются

точками.Имя домена идентифицирует его положение в этой базе данных по отношению к родительскому домену. Корень базы данных DNS управляется центром Internet Network Information Center. Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, а для различных типов организаций используются следующие аббревиатуры:

- com - коммерческие организации (например, microsoft.com);
- edu - образовательные (например, mit.edu);
- gov - правительственные организации (например, nsf.gov);
- org - некоммерческие организации (например, fidonet.org);
- net - организации, поддерживающие сети (например, nsf.net).

Каждый домен DNS администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Каждый домен имеет уникальное имя, а каждый из поддоменов имеет уникальное имя внутри своего домена. Имя домена может содержать до 63 символов. Каждый хост в сети Internet однозначно определяется своим полным доменным именем (fully qualified domain name, FQDN), которое включает имена всех доменов по направлению от хоста к корню.

2.3.2. Протокол DHCP

Протокол DHCP был предложен в 1993 г., его развитием занимается специальная рабочая группа (DHC WG), входящая в состав IETF. Наиболее полное современное описание DHCP содержится в документе RFC 2131 (март 1997 г.), который пришел на смену более ранним редакциям RFC 1531 и 1541. В настоящее время DHCP имеет статус предварительного стандарта.

DHCP появился не на пустом месте - различные схемы управления IP-адресами в сетевой среде предлагались и раньше (см. врезку <Распределение IP-адресов...>). Однако эти схемы имеют по крайней мере один из двух недостатков – не допускают динамического назначения IP-адресов либо позволяют передавать от сервера на станцию-клиент лишь небольшое число параметров конфигурации.

При разработке протокола DHCP преследовалась цель устранить оба ограничения. Требовался механизм, который позволил бы ликвидировать стадию ручного конфигурирования компьютеров, поддерживал многосегментные сети, не требуя наличия DHCP-сервера в каждой подсети, не конфликтовал с существующими сетевыми протоколами и компьютерами, имеющими статичную конфигурацию, был способен взаимодействовать с ретранслирующими агентами протокола BOOTP и обслуживать BOOTP-клиентов, наконец, допускал управление передаваемыми параметрами конфигурации. Что касается более узких задач, то DHCP должен был обеспечивать уникальность сетевых адресов, используемых разными компьютерами сети в данный момент, сохранение прежней конфигурации клиентской станции после перезагрузки клиента или сервера, автоматическое присвоение параметров конфигурации вновь подключенным машинам.

Работа протокола DHCP базируется на классической схеме клиент-сервер. В роли клиентов выступают компьютеры сети, стремящиеся получить IP-адреса в так называемую аренду (lease), а DHCP-серверы выполняют функции диспетчеров, которые выдают адреса, контролируют их использование и сообщают клиентам требуемые параметры конфигурации. Сервер поддерживает пул свободных адресов и, кроме того, ведет собственную регистрационную базу данных. Взаимодействие DHCP-серверов со станциями-клиентами осуществляется путем обмена сообщениями.

Упомянутое выше требование поддержки базовых элементов протокола BOOTP возникло не случайно. DHCP разрабатывался как непосредственное расширение BOOTP и именно в таком качестве воспринимается BOOTP-клиентами. Этому обстоятельству в первую очередь способствует формат

сообщений DHCP, во многом совпадающий с форматом, который применяется протоколом-предшественником и определен в документе RFC 951 (рис. 1).

Сравнивая протоколы BOOTP и DHCP, нельзя не отметить появления в DHCP новых услуг. Во-первых, в этом протоколе предусмотрен механизм автоматической выдачи IP-адресов во временное пользование с возможностью их последующего присвоения новым клиентам. Во-вторых, клиент может получить от сервера все параметры конфигурации, которые ему необходимы для успешного функционирования в IP-сети.

Указанные отличия потребовали частичного расширения формата сообщений. Так, в нем появилось отдельное поле идентификатора клиента, сделана более прозрачной интерпретация адреса сервера (поле siaddr), переменный размер получило поле options, используемое, в частности, для передачи параметров конфигурации (его длина обычно находится в диапазоне 312-576 байт, хотя возможно и дополнительное расширение этого поля за счет полей sname и file).

В роли транспортного протокола для обмена DHCP-сообщениями выступает UDP. При отправке сообщения с клиента на сервер используется 67-й порт DHCP-сервера, при передаче в обратном направлении - 68-й. Эти номера портов, как и схожая структура сообщений, обеспечивают обратную совместимость DHCP с BOOTP. Конкретные процедуры взаимодействия клиентов и серверов BOOTP и DHCP регламентирует документ RFC 1542.

2.3.3. Управление IP-адресами DHCP

Протокол DHCP поддерживает три механизма выделения адресов: автоматический, динамический и ручной. В первом случае клиент получает постоянный IP-адрес, в последнем DHCP используется только для уведомления клиента об адресе, который администратор присвоил ему вручную. Оба эти варианта не таят в себе чего-либо принципиально нового, а вот динамический механизм заслуживает детального рассмотрения.

Выдача адреса в аренду производится по запросу клиента. DHCP-сервер (или группа серверов) гарантирует, что выделенный адрес до истечения срока его аренды не будет выдан другому клиенту; при повторных обращениях сервер старается предложить клиенту адрес, которым тот пользовался ранее. Со своей стороны, клиент может запросить пролонгацию срока аренды адреса либо, наоборот, досрочно отказаться от него. Протоколом предусмотрена также выдача IP-адреса в неограниченное пользование. При острой нехватке адресов сервер может сократить срок аренды адреса по сравнению с запрошенным.

Порядок выдачи нового адреса:

1. Клиент посылает в собственную физическую подсеть широковещательное сообщение DHCPDISCOVER, в котором могут указываться устраивающие клиента IP-адрес и срок его аренды. Если в данной подсети DHCP-сервер отсутствует, сообщение будет передано в другие подсети ретранслирующими агентами протокола BOOTP (они же вернут клиенту ответные сообщения сервера).

2. Любой из DHCP-серверов может ответить на поступившее сообщение DHCPDISCOVER сообщением DHCPOFFER, включив в него доступный IP-адрес (yiaddr) и, если требуется, параметры конфигурации клиента. На этой стадии сервер не обязан резервировать указанный адрес. В принципе, он имеет право предложить его другому клиенту, также отправившему запрос DHCPDISCOVER. Тем не менее спецификации RFC 2131 рекомендуют серверу без необходимости не применять подобную тактику, а кроме того, убедиться (например, выдав эхо-запрос ICMP) в том, что предложенный адрес в текущий момент не используется каким-либо из компьютеров сети.

3. Клиент не обязан реагировать на первое же поступившее предложение. Допускается, чтобы он дождался откликов от нескольких серверов и, остановившись на одном из предложений, отправил в сеть широковещательное сообщение DHCPREQUEST. В нем содержатся идентификатор выбранного сервера и, возможно, желательные значения запрашиваемых параметров конфигурации.

Не исключено, что клиента не устроит ни одно из серверных предложений. Тогда вместо DHCPREQUEST он снова выдаст в сеть запрос DHCPDISCOVER, а серверы так и не узнают, что их предложения отклонены. Именно по этой причине сервер не обязан резервировать помещенный в DHCPOFFER адрес.

Если в процессе ожидания серверных откликов на DHCPDISCOVER достигнут тайм-аут, клиент выдает данное сообщение повторно.

4. Присутствующий в сообщении DHCPREQUEST идентификатор позволяет соответствующему DHCP-серверу убедиться в том, что клиент принял именно его предложение. В ответ сервер отправляет подтверждение DHCPACK, содержащее значения требуемых параметров конфигурации, и производит соответствующую запись в базу данных.

Если к моменту поступления сообщения DHCPREQUEST предложенный адрес уже <ушел> к другому клиенту (например, первая станция слишком долго <размышляла> над поступившими предложениями), сервер отвечает сообщением DHCPNACK.

5. Получив сообщение DHCPACK, клиент обязан убедиться в уникальности IP-адреса (средствами протокола ARP) и зафиксировать суммарный срок его аренды. Последний рассчитывается как время, прошедшее между отправкой сообщения DHCPREQUEST и приемом ответного сообщения DHCPACK, плюс срок аренды, указанный в DHCPACK.

Обнаружив, что адрес уже используется другой станцией, клиент обязан отправить серверу сообщение DHCPDECLINE и не ранее чем через 10 с начать всю процедуру снова. Процесс конфигурирования возобновляется и при получении серверного сообщения DHCPNACK.

При достижении тайм-аута в процессе ожидания серверных откликов на сообщение DHCPREQUEST клиент выдает его повторно.

6. Для досрочного прекращения аренды адреса клиент отправляет серверу сообщение DHCPRELEASE.

Приведенная последовательность действий заметно упрощается, если станция-клиент желает повторно работать с IP-адресом, который когда-то уже

был ей выделен. В этом случае первым отправляемым сообщением является DHCPREQUEST, в котором клиент указывает прежде использовавшийся адрес. В ответ он может получить сообщение DHCPACK или DHCPNACK (если адрес занят либо клиентский запрос является некорректным, например из-за перемещения клиента в другую подсеть). Обязанность проверить уникальность IP-адреса опять-таки возлагается на клиента.

Выбор адреса DHCP-сервером. Если на момент получения запроса DHCPDISCOVER сервер не располагает свободными IP-адресами, он может направить уведомление о возникшей проблеме администратору. В противном случае при выборе адреса обычно применяется следующий алгоритм. Клиенту выделяется адрес, записанный за ним в данный момент. Если это невозможно, сервер предложит адрес, которым пользовался клиент до окончания срока последней аренды (при условии, что данный адрес свободен), либо адрес, запрошенный самим клиентом при помощи соответствующей опции (опять же, если адрес не занят). Наконец, в том случае, когда все предыдущие варианты не проходят, новый адрес выбирается из пула доступных адресов с учетом подсети, из которой поступил клиентский запрос.

Заметим, что исходя из определенной сетевым администратором политики сервер может выдать клиенту адрес, отличающийся от запрошенного (даже при доступности последнего), вообще отказать в предоставлении адреса или предложить адрес, относящийся к другой подсети. Более того, DHCP-сервер вообще не обязан реагировать на каждый поступивший запрос DHCPDISCOVER. Это предоставляет администратору возможность контролировать доступ к сети, например разрешив серверу отвечать только тем клиентам, которые предварительно зарегистрировались с помощью специальной процедуры.

Истечение срока аренды. По мере того, как срок аренды подходит к концу, клиент может завершить работу с данным адресом, отправив на DHCP-сервер сообщение DHCPRELEASE, либо заблаговременно запросить продление срока аренды. В первом случае возвращение в сеть потребует выполнения всей

процедуры инициализации заново. Во втором - станция продолжит функционировать в сети без видимого замедления работы пользовательских приложений.

При пролонгировании аренды клиент проходит два состояния - обновления адреса (RENEWING) и обновления конфигурации (REBINDING). Первое наступает примерно на половине срока аренды адреса (так называемый момент T1), второе - по истечении приблизительно 7/8 полного времени аренды (момент T2); для рассинхронизации процессов реконфигурирования разных клиентов значения этих временных меток рандомизируются с помощью случайной добавки.

В момент T1 клиент оправляет DHCP-серверу, выдавшему адрес, сообщение DHCPREQUEST с просьбой продлить срок аренды. Получив положительный ответ (DHCPACK), клиент пересчитывает срок аренды и продолжает работу в обычном режиме. Клиент ожидает прихода ответа от сервера в течение (T2 - t)/2 с (при условии, что это значение не меньше 60 с), где t - время отсылки последнего сообщения DHCPREQUEST, после чего отправляет данное сообщение повторно.

Если ответ от сервера не поступил к моменту T2, клиент переходит в состояние REBINDING и передает уже широковещательное сообщение DHCPREQUEST со своим текущим сетевым адресом. В этом случае моменты повторных выдач запросов DHCPREQUEST рассчитываются аналогично предыдущему случаю, только вместо T2 фигурирует время окончания срока аренды.

Не исключено, однако, что ответ DHCPACK не придет до окончания срока аренды. Тогда клиент обязан немедленно прекратить выполнение любых сетевых операций и заново начать процесс инициализации. Если запоздавший ответ DHCPACK все-таки поступит, клиенту рекомендуется сразу же возобновить работу под прежним адресом.

2.3.4. Параметры конфигурации DHCP-сервера

Хранение параметров сетевой конфигурации станций-клиентов является второй услугой, предоставляемой DHCP-сервером. В создаваемой базе данных на каждого клиента заводится отдельная запись с уникальным ключом-идентификатором и строкой конфигурационных параметров.

Роль идентификатора может играть пара <номер подсети IP, аппаратный адрес>, которая позволит использовать аппаратный адрес сразу в нескольких подсетях, либо пара <номер подсети IP, имя хост-компьютера>, позволяющая серверу взаимодействовать с клиентом, перемещенным в другую подсеть.

Что касается собственно параметров конфигурации, то их набор, поддерживаемый протоколом DHCP, определен в спецификациях RFC 1122, 1123, 1196 и 1256. В него входят выданный адрес, срок его аренды, назначавшиеся ранее адреса, а также максимальный размер реассемблируемого пакета, перечень фильтров для нелокальной маршрутизации от источника, адрес, используемый в широковещательных пакетах, параметры статических маршрутов и т.д. Впрочем, из всей совокупности допустимых параметров (а их более 30) в процессе инициализации могут передаваться только те, которые действительно необходимы для работы клиента либо определяются спецификой конкретной подсети.

Редукция объема передаваемых сведений о конфигурации достигается двумя способами. Во-первых, для большей части параметров в упомянутых выше документах RFC определены значения, принимаемые по умолчанию. Клиент будет использовать их, если в сообщении, поступившем от сервера, какие-то параметры опущены. Во-вторых, отправляя сообщение DHCPDISCOVER или DHCPREQUEST, клиентская станция может явно указать в нем параметры, значения которых она хотела бы получить.

Очевидно, что в обоих случаях передача параметров конфигурации осуществляется в ходе основной процедуры выделения IP-адреса. Возможен, однако, случай, когда клиент уже имеет IP-адрес (например, он был задан

вручную). Тогда он может выдать сообщение DHCPINFORM*, содержащее уже имеющийся адрес и запрос об отдельных параметрах конфигурации. Получив это сообщение, DHCP-сервер проверяет правильность адреса клиента (но не наличие аренды) и направляет ему сообщение DHCPACK с требуемыми параметрами конфигурации.

Отметим одно логическое противоречие, с которым связано применение протокола DHCP. Алгоритм выделения IP-адреса компьютеру сети предполагает, что установленное на нем программное обеспечение TCP/IP в состоянии воспринимать адресованные ему посредством <аппаратного> адреса IP-пакеты и транслировать их на IP-уровень еще до того, как станция получит свой IP-адрес, а сами средства TCP/IP будут полностью сконфигурированы. Такая возможность, очевидно, существует не всегда. Для работы с клиентами, не способными корректно обрабатывать одноадресные IP-дейтаграммы, используется поле flags. Такие клиенты должны установить первый бит данного поля в единичное значение, тем самым указав серверу на необходимость отправки в соответствующую подсеть только широковещательных сообщений.

2.4. Служба каталогов Active Directory

2.4.1. Понятие и назначение службы каталогов Active Directory

Современные сети часто состоят из множества различных программных платформ, большого разнообразия оборудования и программного обеспечения. Пользователи зачастую вынуждены запоминать большое количество паролей для доступа к различным сетевым ресурсам. Права доступа могут быть различными для одного и того же сотрудника в зависимости от того, с какими ресурсами он работает. Все это множество взаимосвязей требует от администратора и пользователя огромного количества времени на анализ, запоминание и обучение.

Решение проблемы управления такой разнородной сетью было найдено с разработкой службы каталога. Службы каталога предоставляют возможности

управления любыми ресурсами и сервисами из любой точки независимо от размеров сети, используемых операционных систем и сложности оборудования. Информация о пользователе, заносится единожды в службу каталога, и после этого становится доступной в пределах всей сети. Адреса электронной почты, принадлежность к группам, необходимые права доступа и учетные записи для работы с различными операционными системами — все это создается и поддерживается в актуальном виде автоматически. Любые изменения, занесенные в службу каталога администратором, сразу обновляются по всей сети. Администраторам уже не нужно беспокоиться об уволенных сотрудниках — просто удалив учетную запись пользователя из службы каталога, он сможет гарантировать автоматическое удаление всех прав доступа на ресурсы сети, предоставленные ранее этому сотруднику.

В настоящее время большинство служб каталогов различных фирм базируются на стандарте X.500. Для доступа к информации, хранящейся в службах каталогов, обычно используется протокол Lightweight Directory Access Protocol (LDAP). В связи со стремительным развитием сетей TCP/IP, протокол LDAP становится стандартом для служб каталогов и приложений, ориентированных на использование службы каталога.

Служба каталогов Active Directory является основой логической структуры корпоративных сетей, базирующихся на системе Windows. Термин " Каталог " в самом широком смысле означает " Справочник ", а служба каталогов корпоративной сети — это централизованный корпоративный справочник. Корпоративный каталог может содержать информацию об объектах различных типов. Служба каталогов Active Directory содержит в первую очередь объекты, на которых базируется система безопасности сетей Windows, — учетные записи пользователей, групп и компьютеров. Учетные записи организованы в логические структуры: домен, дерево, лес, организационные подразделения.

2.4.2. Назначение службы каталогов Active Directory

Каталог (справочник) может хранить различную информацию, относящуюся к пользователям, группам, компьютерам, сетевым принтерам, общим файловым ресурсам и так далее — будем называть все это объектами. Каталог хранит также информацию о самом объекте, или его свойства, называемые атрибутами. Например, атрибутами, хранимыми в каталоге о пользователе, может быть имя его руководителя, номер телефона, адрес, имя для входа в систему, пароль, группы, в которые он входит, и многое другое. Для того чтобы сделать хранилище каталога полезным для пользователей, должны существовать службы, которые будут взаимодействовать с каталогом. Например, можно использовать каталог как хранилище информации, по которой можно аутентифицировать пользователя, или как место, куда можно послать запрос для того, чтобы найти информацию об объекте.

Active Directory отвечает не только за создание и организацию этих небольших объектов, но также и за большие объекты, такие как домены, OU (организационные подразделения) и сайты.

Об основных терминах, используемых в контексте службы каталогов Active Directory, читайте ниже.

Служба каталогов Active Directory (сокращенно — AD) обеспечивает эффективную работу сложной корпоративной среды, предоставляя следующие возможности:

1. Единая регистрация в сети; Пользователи могут регистрироваться в сети с одним именем и паролем и получать при этом доступ ко всем сетевым ресурсам и службам (службы сетевой инфраструктуры, службы файлов и печати, серверы приложений и баз данных и т. д.).

2. Безопасность информации. Средства аутентификации и управления доступом к ресурсам, встроенные в службу Active Directory, обеспечивают централизованную защиту сети.

3. Централизованное управление. Администраторы могут централизованно управлять всеми корпоративными ресурсами.

4. Администрирование с использованием групповых политик. При загрузке компьютера или регистрации пользователя в системе выполняются требования групповых политик; их настройки хранятся в объектах групповых политик (GPO) и применяются ко всем учетным записям пользователей и компьютеров, расположенных в сайтах, доменах или организационных подразделениях.

5. Интеграция с DNS. Функционирование служб каталогов полностью зависит от работы службы DNS. В свою очередь серверы DNS могут хранить информацию о зонах в базе данных Active Directory.

6. Расширяемость каталога. Администраторы могут добавлять в схему каталога новые классы объектов или добавлять новые атрибуты к существующим классам.

7. Масштабируемость. Служба Active Directory может охватывать как один домен, так и множество доменов, объединенных в дерево доменов, а из нескольких деревьев доменов может быть построен лес.

8. Репликация информации. В службе Active Directory используется репликация служебной информации в схеме со многими ведущими (multi-master), что позволяет модифицировать БД Active Directory на любом контроллере домена. Наличие в домене нескольких контроллеров обеспечивает отказоустойчивость и возможность распределения сетевой нагрузки.

9. Гибкость запросов к каталогу. БД Active Directory может использоваться для быстрого поиска любого объекта AD, используя его свойства (например, имя пользователя или адрес его электронной почты, тип принтера или его местоположение и т. п.).

10. Стандартные интерфейсы программирования. Для разработчиков программного обеспечения служба каталогов предоставляет доступ ко всем возможностям (средствам) каталога и поддерживает принятые стандарты и интерфейсы программирования (API).

В Active Directory может быть создан широкий круг различных объектов. Объект представляет собой уникальную сущность внутри Каталога и обычно обладает многими атрибутами, которые помогают описывать и распознавать его. Учетная запись пользователя является примером объекта. Этот тип объекта может иметь множество атрибутов, таких как имя, фамилия, пароль, номер телефона, адрес и многие другие. Таким же образом общий принтер тоже может быть объектом в Active Directory и его атрибутами являются его имя, местоположение и т.д. Атрибуты объекта не только помогают определить объект, но также позволяют вам искать объекты внутри Каталога.

3. АДМИНИСТРИРОВАНИЕ ЭЛЕМЕНТОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ РАБОЧИХ СТАНЦИЙ НА ОСНОВЕ ОПЕРАЦИОННЫХ СИСТЕМ WINDOWS

3.1. Средства защиты информации в операционных системах

3.1.1. Архитектура подсистемы безопасности операционной системы Windows

Основные компоненты подсистемы безопасности операционной системы Microsoft Windows представлены на рис. 3.1 (ID и P — имя и аутентификатор пользователя, H — функция хеширования, SID — идентификатор безопасности учетной записи пользователя, AT— маркер безопасности, SD — дескриптор безопасности объекта доступа, R — результат проверки прав доступа процесса пользователя к объекту).

Ядром подсистемы безопасности является локальная служба безопасности (Local Security Authority — LSA), размещающаяся в файле lsass.exe. После загрузки операционной системы автоматически запускается процесс входа (winlogon.exe), который остается активным до перезагрузки операционной системы или выключения питания компьютера, а его аварийное завершение приводит к аварийному завершению и всей операционной системы. Этим обеспечивается практическая невозможность подмены процесса входа при функционировании системы.

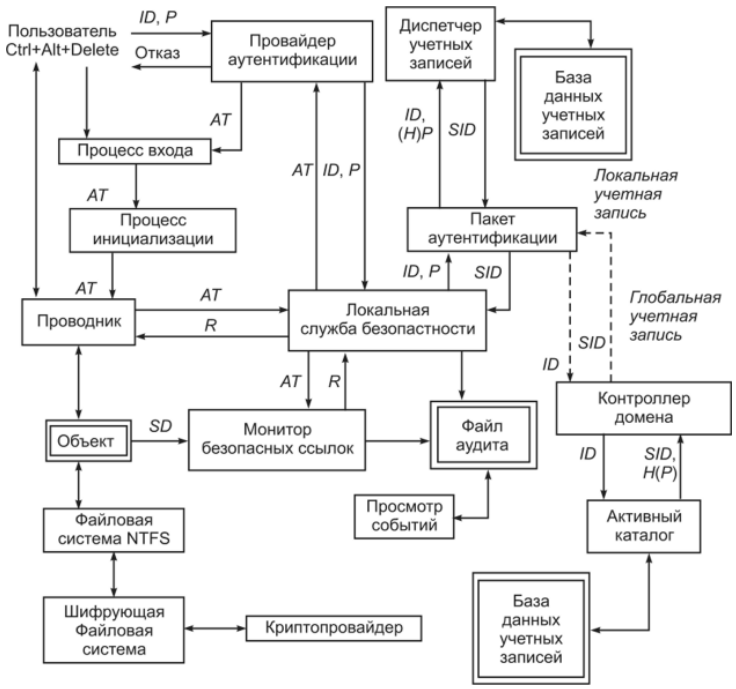

Рисунок 3.1. Подсистема безопасности ОС Windows.

Процесс входа в систему связывается с поставщиком динамической библиотеки функций (DLL), чтобы получить от пользователя логическое имя (ID) и информацию для аутентификации (F). Поставщик проверки подлинности по умолчанию находится в библиотеке динамической компоновки для графической идентификации и проверки подлинности (msgina.dll) и использует пароли пользователей, введенные с клавиатуры или считанные со смарт-карты, в качестве информации для проверки подлинности. Вы можете использовать услуги других поставщиков аутентификации (например, с помощью биометрической аутентификации). В этом случае интерфейс функций, предоставляемых поставщиком аутентификации, должен соответствовать определениям в файле winwlx.h. Путь к провайдеру аутентификации, используемому процессом входа в систему, записан в разделе реестра *HKEY_LOCAL_MACHINESoftwareMicrosoftWindowsNTCurrentVersion WinLogon* в качестве значения параметра GinaDLL. В таблице 3.1 перечислены функции, экспортируемые поставщиком аутентификации.

Таблица 3.1. Функции, экспортируемые библиотекой GINA

Имя функции	**Описание**
WIxActivateUserShell	Активизация сценария, выполняемого при успешном входе пользователя
WlxDisplayLockedNotice	Разрешение отображения информации о блокировании компьютера
WlxDisplaySASNotice	Вызов при отсутствии входа пользователя
WlxDisplayStatusMessage	Вывод сообщения пользователю
WlxGetConsoleSwitchCredentials	Получение информации о результате попытки входа пользователя
WlxGetStatusMessage	Получение текста сообщения пользователю
Wlxlnitialize	Инициализация библиотеки GINA
WIxIsLockOk	Проверка возможности блокировки компьютера
WlxIslogoffOk	Проверка возможности выхода пользователя
WIxLoggedOnSAS	Нажатие Ctrl+Alt+Delete после входа пользователя и отсутствии блокировки компьютера
WIxLoggedOutSAS	Нажатие Ctrl+Alt+Delete перед входом пользователя
WIxLogoff	Извещение о выходе пользователя
WlxNegotiate	Спецификация текущей версии процесса входа
WlxNetworkProviderLoad	Получение результата аутентификации глобальной учетной записи
WlxRemoveStatusMessage	Указание о прекращении вывода сообщения пользователю

WIxScreensaverNotify	Разрешение начала интерактивного взаимодействия с хранителем экрана
WlxShutdown	Извещение о завершении работы системы
WIxStartApplication	Вызов приложения от имени пользователя
WlxWkstaLockedSAS	Нажатие Ctrl+Alt+Delete после блокировки компьютера

Процесс входа использует сообщение Windows WLX WM SAS для извещения библиотеки GINA о различных событиях (табл. 3.2).

Таблица 3.2. Сообщения для библиотеки GINA

Значение параметра wParam сообщения	**Описание**
WLX_SAS_TYPE_CTRL_ALT_DEL	Нажатие Ctrl+Alt+Delete
WLX_SAS_TYPE_SC_I NSERT	Вставка смарт-карты
WLX_SAS_TYPE_SC_REMOVE	Извлечение смарт-карты
WLX_SAS_TYPE_USER_LOGOFF	Попытка входа пользователя
WLX_SAS_TYPE_SCRNSVR_TIMEOUT	Запуск хранителя экрана
W LX_SAS_TY P E_T 1M E 0 UT	Отсутствие попытки входа пользователя за установленный период времени

Для организации диалога с пользователем в библиотеке GINA должны использоваться функции WlxMessageBox, WlxDialogBox, WlxDialogBoxIndirect, WlxDialogBoxParam и Wlx- DialogBoxIndirectParam вместо соответствующих функций из набора Windows API (без префикса Wlx) для гарантии контроля процесса входа и провайдера аутентификации над стандартными диалогами

Windows. Ошибка при выполнении версий процесса входа для этих функций может быть результатом попытки получения доступа к системе неавторизованного пользователя.

Другие функции, которые могут вызываться функциями библиотеки GINA, приведены в табл. 3.3.

Таблица 3.3. Функции, вызываемые провайдером аутентификации

Имя функции	**Описание**
WIxAssignShellProtection	Запрос защиты сценария, выполняемого при входе пользователя
WlxChangePasswordNotify	Посылка извещения о смене пароля пользователем
WlxChangePasswordNotifyEx	
WlxCloseUserDesktop	Закрытие рабочего стола пользователя
WlxCreatellserDesktop	Создание рабочего стола пользователя
WlxDisconnect	Разрыв соединения со службой терминалов
WlxGetOption	Определение текущего значения заданной опции (например, возможности входа со смарт-картой)
WlxGetSourceDesktop	Определение имени и дескриптора рабочего стола, который был активным до переключения на рабочий стол процесса входа
WlxQueryClientCredentials	Запрос аутентификатора удаленного клиента службы терминалов
WlxQuerylnetConnectorCredentials	
WlxQueryConsoleSwitchCredentials	Запрос аутентификатора, переданного от рабочего стола временного сеанса

WlxQueryTerminalServicesData	Запрос у службы терминалов конфигурационной информации пользователя
WIxSasNotify	Извещение процесса входа о нажатии Ctrl+Alt+Delete
WlxSetContextPointer	Определение указателя на контекст, передаваемый процессу входа в качестве первого параметра при вызове функций GINA
WlxSetOption	Задание значения указанной опции
WIxSetReturnDesktop	Определение рабочего стола процесса входа, который будет активизирован после нажатия Ctrl+Alt+Delete
WIxSetTimeout	Изменение времени задержки, связанной с диалогом аутентификации
WIxSwitchDesktopToUser	Переключение на рабочий стол приложений
WIxSwitchDesktopToWinlogon	Переключение на рабочий стол процесса входа
WlxWin31 Migrate	Завершение установки клиента службы терминалов
WIxUseCtrlAltDel	Сообщение процессу входа об использовании Ctrl+Alt+Delete в качестве стандартной комбинации клавиш

Рабочий стол в операционной системе Windows представляет собой набор окон, которые одновременно отображаются на экране. Только процессы, окна которых находятся на одном рабочем столе, могут взаимодействовать друг с другом с помощью графического интерфейса пользователя (GUI) Windows. Все рабочие станции включены в процесс входа в систему.

После успешного входа пользователя в систему приложения, которые он запускает, запускаются на рабочем столе приложения (рабочем столе пользователя). Когда заставка запущена, рабочий стол заставки активируется.

Как владелец всех трех рабочих станций, процесс входа может переключаться между ними. Когда вызываются функции GINA, процесс входа в систему указывает, какой рабочий стол является текущим (видимым) рабочим столом. Никакой другой процесс, включая вкладку программного обеспечения, введенную злоумышленником для перехвата паролей, не может получить доступ к процессу входа в систему или к рабочему столу заставки. Таким образом, запрос на вход пользователя в систему, отображаемый на этой вкладке, может быть только на рабочем столе приложения, где связаны все программы, запускаемые пользователем.

Сообщение о нажатии Ctrl+Alt+Del отправляется только процессу входа в систему, который, помните, остается активным до тех пор, пока операционная система не будет выключена (перезапущена) или отключено питание. Для всех остальных процессов (особенно для всех приложений, запускаемых пользователем) нажатие этой комбинации клавиш совершенно невидимо.

После загрузки системы на экране компьютера сначала отображается рабочий стол процесса входа в систему. Однако пользователь вводит логин и пароль не сразу, а только после нажатия Ctrl+Alt+Del. Когда пользователь выходит из системы, также активируется информационная панель процесса входа в систему, и новый пользователь может ввести пароль для входа только после нажатия Ctrl+Alt+Del.

Если в системе есть программная закладка для перехвата паролей, то для выполнения этого перехвата в этой вредоносной программе необходимо обработать пользователя нажатием комбинации клавиш Ctrl+Alt+Delete. В противном случае, когда пользователь нажимает эту комбинацию клавиш, процесс входа в систему переходит на рабочий стол. Рабочий стол приложения будет неактивным, и закладка просто не сможет ничего перехватить - сообщения о пользователе при нажатии клавиш будут попадать на другой рабочий стол.

Однако для всех приложений тот факт, что пользователь нажимает комбинацию Ctrl+Alt+Delete, всегда остается незамеченным. Следовательно, пароль, введенный пользователем, будет принят не программной закладкой, а поставщиком аутентификации.

Очевидно, программная закладка может имитировать не первое приглашение операционной системы, в которой пользователю предлагают нажать Ctrl+Alt+Delete для запуска задания, а приглашение, которое отображается после того, как пользователь нажимает эту комбинацию.

Однако при нормальных обстоятельствах (при отсутствии закладки) это второе приглашение автоматически отменяется через довольно короткий промежуток времени (от 30 секунд до 1 минуты, в зависимости от версии Windows). Если второе приглашение продолжает появляться на экране компьютера в течение длительного времени, этот факт должен насторожить пользователя. Кроме того, опыт показывает, что у пользователей с достаточным опытом работы с Windows появляется привычка запускать систему, нажимая Ctrl+Alt+Delete, независимо от того, что отображается на экране. Поэтому защиту операционной системы Windows от подобных программных ошибок можно считать очень надежной.

Имя и пароль, предоставленные пользователем, передаются поставщиком аутентификации службе LSA, которая обращается к пакету аутентификации (динамически подключаемая библиотека функций) для аутентификации введенного имени пользователя. Пакет проверки подлинности MSV10 (из файла msvl_0.dll) используется для проверки подлинности локальной учетной записи, зарегистрированной на компьютере, на котором пользователь осуществляет попытку входа.

В этом случае пакет аутентификации вычисляет хеш-значение пароля H (P) и связывается с менеджером учетных записей безопасности (SAM), чтобы убедиться, что введенный пароль правильный и что пользователь с введенным логическим именем может начать работу в системе (не истек ли срок действия пароля, учетная запись пользователя заблокирована и т. д.).

Менеджер учетных записей обращается к базе данных учетных записей (SAM) для получения информации из учетной записи пользователя с введенным логическим именем. База данных учетной записи содержится в разделе реестра HKEYLOC ALM ACHINESAM (в файле WindowsSystem32 ConfigSAM). Администратор не может получить доступ к базе данных SAM с помощью стандартных средств операционной системы для чтения или изменения (файл с базой данных открывается ядром операционной системы во время процесса запуска в монопольном режиме). Для редактирования базы данных SAM используются специальные функции ряда Windows API и специального системного программного обеспечения (оснастки «Управление компьютером» и «Локальная политика безопасности»).

Пароль пользователя в базе данных SAM хранится в виде двух хеш-значений длиной 128 бит каждое. Первое из этих хеш-значений генерируется с использованием алгоритма Windows NT.:

1. Строка символов пароля P усекается до 14 знаков (при необходимости) и преобразуется в кодировку Unicode, в которой каждый символ представляется двумя байтами.

2. Вычисляется хеш-значение преобразованного пароля *H(P)* длиной 128 бит (используется функция хеширования MD4).

3. Полученное хеш-значение зашифровывается по алгоритму DES с помощью ключа, выводимого из относительного номера учетной записи пользователя, $E_{RID}(H(P))$.

4. Полученный результат шифрования записывается в базу данных учетных записей.

Второе хеш-значение пароля пользователя вычисляется по алгоритму LAN Manager:

1. Все буквенные символы (латинского алфавита) строки пароля *P* преобразуются к верхнему регистру.

2. Строка символов пароля дополняется нулями, если она короче 14 байтов, и делится на две семибайтовые половины P_1 и P_2.

3. Каждое из значений P_1 и P_2 используется в качестве ключа для шифрования по алгоритму DES магической строки *M = «KGS!@#$%»,* в результате которого получаются два значения из 64 бит каждое — H_1, = $E_{P1}(M)$ и $H_2 = E_{P2}(M)$.

4. Выполняется шифрование по алгоритму DES на ключе, равном относительному номеру учетной записи, результата сцепления H_1 и H_2 — $E_{RID}(H_1 || H_2)$.

5. Полученный результат шифрования помещается в базу данных SAM.

Алгоритм получения хеш-значения LAN Manager гораздо менее безопасен (мощность алфавита, из которого выбираются символы пароля, искусственно занижена, а разделение пароля на две половины позволяет легче угадать, не превышает ли длина пароля семь символов, поскольку результат шифрования магической строки на нулевом ключе злоумышленнику заранее известен). Использование «более слабого» алгоритма LAN Manager объясняется необходимостью обеспечить совместимость с более ранними версиями операционной системы Windows, которые могут быть установлены на рабочие станции в локальной сети организации.

Другой недостаток хранения паролей пользователей в защищенных версиях Windows заключается в том, что без учета окончательного шифрования в ключе, равном относительному номеру учетной записи, в обоих алгоритмах один и тот же пароль, случайно выбранный разными пользователями, будет соответствовать одному и тому же хэш-значению. Это создает теоретическую возможность использования полномочий одного пользователя другим, узнавшим об этом факте. Относительный номер учетной записи может быть легко получен злоумышленником (например, учетная запись внутреннего администратора всегда имеет относительный номер 500).

Для устранения такой уязвимости достаточно будет вычислить хеш-значение пароля, используя дополнительное случайное число, генерируемое при регистрации пользователя.

Несмотря на то, что злоумышленнику практически невозможно получить доступ к базе данных SAM стандартными средствами Windows, он все же может, например, скопировать файл, содержащий его, на подготовленный носитель после загрузки другой операционной системы на компьютере с подготовленного съемного диска. (предотвратить это можно только с помощью организационных мер). Затем информация из копии файла SAM восстанавливается на компьютере злоумышленника, который теперь может получить доступ к базе данных учетной записи с помощью специального программного обеспечения.

Следующим шагом злоумышленника будет программный выбор паролей для привилегированных пользователей, которые он получил, и что он сможет осуществить несанкционированный доступ к информации. Чтобы предотвратить эту угрозу, администратор безопасности Windows может дополнительно зашифровать хэши паролей пользователей в базе данных SAM с помощью системной программы syskey. Программа syskey шифрует хэши паролей с помощью 128-битного первичного ключа, который хранится в реестре, а также зашифрован на системном ключе.

После запуска программы syskey администратор должен выбрать способ хранения системного ключа длиной также 128 бит, который будет использован для шифрования первичного ключа:

– в системном реестре (преимущество этого варианта — отсутствие необходимости присутствия привилегированного пользователя при загрузке или перезагрузке операционной системы, а недостаток — наименьшая защищенность хранения системного ключа);

– в файле startup.key (длиной 16 байт) в корневом каталоге специальной дискеты (в этом случае придется отдельно позаботиться о защищенном хранении этой дискеты и ее резервной копии);

– без физического сохранения системного ключа, который будет генерироваться из специальной парольной фразы длиной не менее 12 символов.

Во втором и третьем вариантах при загрузке операционной системы перед появлением приглашения следует нажать комбинацию клавиш Ctrl+Alt+ Delete,

появится диалоговое окно с приглашением вставить дискету с системным ключом шифрования или ввести парольную фразу.

Другим вариантом усложнения задачи подбора пароля пользователя по его похищенному хеш-значению является включение параметра безопасности «Сетевая безопасность: не хранить хеш-значений Lan Manager при следующей смене пароля». Но возможность подбора пароля, хешированного по более стойкому алгоритму NTLM, у нарушителя все равно остается.

Если пользователь пытается войти в систему под именем глобальной учетной записи, зарегистрированной в домене (объединении компьютеров, разделяющих общую политику безопасности и информацию об учетных записях пользователей), то пакет аутентификации MSV10 отправляет логическое имя и хеш-значение пароля пользователя сервису NetLogon для аутентификации пользователя на контроллере домена (сервере аутентификации). Далее выполняется аутентификация пользователя по протоколам LM, NTLM и NTLM 2.

Но основным пакетом аутентификации при входе пользователя под именем глобальной учетной записи в современных версиях ОС Windows является пакет аутентификации Kerberos SSP/AP (Security Support Provider/Authentication Packages, провайдер поддержки безопасности/пакет аутентификации), использующий более защищенный протокол непрямой аутентификации Kerberos.

Если результат аутентификации пользователя, проведенный с помощью введенных им данных, отрицателен или работа пользователя в системе невозможна (пользователь с такой учетной записью не зарегистрирован, учетная запись пользователя заблокирована или отключена, пользователь с такой учетной записью не может входить в домен в это время или на этом компьютере), то пакет аутентификации возвращает LSA код ошибки, который локальная служба безопасности передает процессу входа для выдачи пользователю сообщения об отказе в доступе к системе.

Если проверка подтвердила подлинность пользователя и отсутствие препятствий для начала его работы в КС, пакет аутентификации получает

уникальный идентификатор безопасности пользователя SID (security identifier), который затем передается в LSA.

Идентификатор безопасности представляет собой структуру переменной длины, которая однозначно определяет пользователя или группу и сохраняется в его учетной записи в регистрационной базе данных. В SID содержится следующая информация:

– идентификатор авторизации (48 бит), состоящий из идентификатора домена КС и относительного номера учетной записи (RID) в его регистрационной базе данных (32 бита);

– уровень пересмотра;

– переменное количество идентификаторов субавторизации.

Для записи SID применяется специальная нотация:

S-R-I-S-S... (первая S определяет эту запись как SID, R задает уровень пересмотра, I — идентификатор авторизации, вторая и последующие буквы S — идентификаторы субавторизации), например, S-1-4138-86. Некоторые значения SID являются предопределенными в системе, например: S-1-0-0 (группа без пользова- телей-членов), S-1-1-0 (группа, включающая всех пользователей), S-1-2-0 (группа, включающая пользователей, которые входят в КС локально), S-1-3-0 (пользователь, являющийся создателем-вла- дельцем нового объекта в КС), S-1-3-1 (первичная группа пользователя, являющегося создателем-владельцем нового объекта), S-1-5-1 (группа, включающая пользователей, осуществляющих удаленный доступ к КС через модем), S-1-5-2 (группа, включающая пользователей, осуществляющих доступ по сети), S-1-5-18 (псевдопользователь System — операционная система) и др.

Получив идентификатор безопасности пользователя, локальная служба безопасности LSA создает для него маркер доступа AT (access token), который идентифицирует пользователя во всех его действиях с объектами КС. В маркере доступа содержится следующая информация:

– SID пользователя;

– идентификаторы безопасности его групп;

– права (привилегии) пользователя в системе;

– идентификаторы безопасности пользователя и его первичной группы, которые будут использованы при создании пользователем новых объектов в КС;

– дискреционный список контроля доступа по умолчанию для вновь создаваемого объекта;

– источник выдачи маркера доступа;

– тип маркера доступа — первичный (созданный при входе пользователя в КС) или используемый для олицетворения (impersonation);

– текущий уровень олицетворения и др.

В маркер доступа не помещаются сведения о правах входа пользователей в КС, которые проверяются пакетом аутентификации на этапе их авторизации в системе:

– локальный вход в систему;

– доступ к компьютеру из сети;

– вход в качестве пакетного задания;

– вход в качестве службы;

– вход в систему через службу терминалов.

При взаимодействии клиентов и серверов в операционной системе Windows может использоваться механизм олицетворения. В этом случае при необходимости обратиться с запросом к серверу в процессе клиента создается поток (thread), которому назначается маркер доступа пользователя, инициировавшего этот запрос. Этот механизм позволяет серверу действовать от лица клиента при доступе к объектам, к которым сам сервер не имеет доступа.

Созданный LSA маркер доступа AT передается процессу входа, который с помощью провайдера аутентификации завершает процесс авторизации пользователя в КС, запуская процесс его инициализации (userinit.exe) и передавая ему AT. Процесс инициализации на основе содержащегося в AT идентификатора безопасности пользователя загружает из реестра Windows его локальный профиль или из службы активного каталога на контроллере домена его перемещаемый профиль и загружает программную оболочку — проводник

Windows (explorer.exe), передавая ему маркер доступа и профиль пользователя. После этого процесс инициализации завершает свою работу.

При запуске в сеансе пользователя любой программы создаваемому процессу передается копия маркера безопасности пользователя, что обеспечивает передачу процессу и всех прав пользователя в системе. Для запуска приложения с правами другого пользователя, если известны его имя и пароль, предназначена команда контекстного меню файла приложения «Запуск от имени». Эту возможность следует использовать для сокращения времени работы в системе с правами привилегированного пользователя.

3.1.2. Разграничение прав пользователей операционной системы Windows

Для управления списками пользователей и групп в КС, назначения им полномочий, определения параметров политики безопасности в операционных системах Windows администратором используются оснастки «Управление компьютером» и «Локальная политика безопасности» (для локальных учетных записей), «Active Directory — пользователи и компьютеры» и «Политика безопасности домена» (для глобальных учетных записей), а также оснастка «Редактор объектов групповой политики». Эти системные программы могут быть вызваны из консоли управления Microsoft (Microsoft Management Console, mmc.exe) или непосредственно по их именам (например, gpedit.msc для оснастки «Редактор объектов групповой политики»).

На рис. 3.2 и 3.3 приведены примеры определения свойств учетной записи пользователя, а также свойств и состава группы пользователей. Для учетной записи пользователя, помимо его логического имени для входа в систему и полного имени и (или) описания, задаются следующие характеристики:

– признак необходимости смены пароля при следующем входе в систему (устанавливается, как правило, при создании учетной записи администратором или задании для пользователя нового пароля);

– признак запрещения смены пароля пользователем (может устанавливаться для «обобщенных» учетных записей, объединяющих нескольких пользователей с минимальными правами в системе, например студентов);

– признак неограниченности срока действия пароля (должен применяться в исключительных случаях, так как отрицательно влияет на безопасность информации в КС);

– признак отключения учетной записи (используется при необходимости временно запретить пользователю возможность доступа к КС);

– признак блокировки учетной записи (устанавливается автоматически при превышении максимального числа неудачных попыток входа в систему и снимается администратором «вручную» или по истечении заданного периода времени).

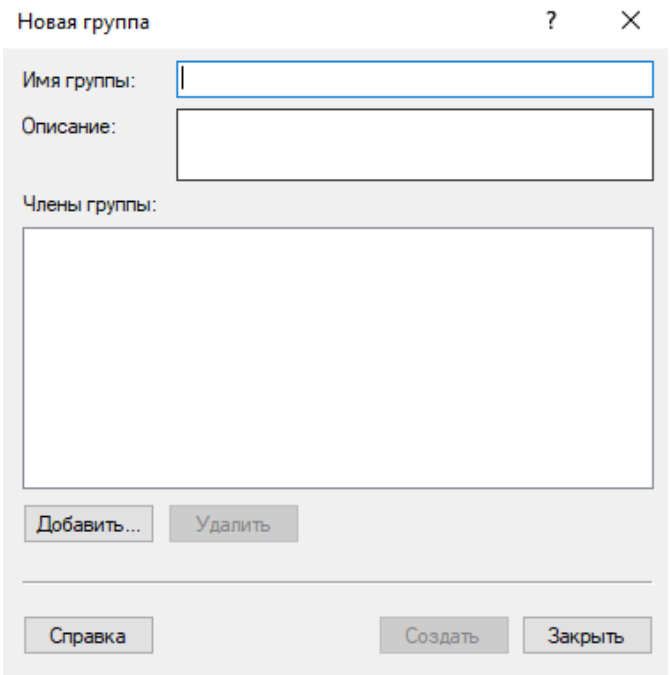

Рисунок 3.2. Определение свойств учетной записи новой группы.

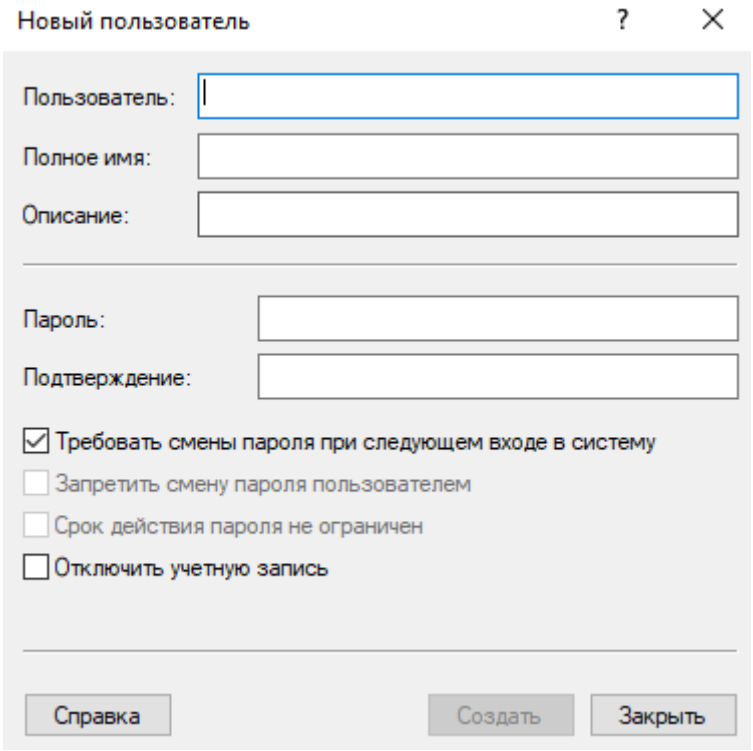

Рисунок 3.3. Определение свойств учетной записи нового пользователя.

У администратора имеется возможность задания пользователю нового пароля (с помощью соответствующей команды контекстного меню учетной записи пользователя), однако это право следует применять осторожно (если пользователь забыл свой пароль). Задание нового пароля может привести, в частности, к невозможности для пользователя работать с зашифрованными им средствами шифрующей файловой системы документами и папками.

В операционный системе Windows предопределены несколько псевдопользователей, например «Система» (SYSTEM) или «Создатель—владелец» объекта КС (CREATOROWNER). Автоматически создаются учетные записи «Администратор» (Administrator) и «Гость» (Guest). Для запрещения анонимного входа в КС учетная запись «Гость» должна быть отключена (что и делается по умолчанию при установке операционной системы). Для предотвращения угрозы компрометации пароля встроенной учетной записи «Администратор», для которой нарушителю известны не только логическое имя, но и относительный номер RID, эта учетная запись также должна быть

переименована (параметр безопасности «Учетные записи: Переименование учетной записи администратора») или отключена (параметр безопасности «Учетные записи: Состояние учетной записи «Администратор») после создания другой учетной записи с полномочиями администратора КС.

В свойствах группы указываются ее имя, описание и список входящих в ее состав пользователей и других групп. К предопределенным группам относятся «Все» (Everyone, включает пользователей и псевдопользователей КС), «Интерактивные» (Interactive), «Входящие по сети» (Network), «Удаленные» (Dial_Up), «Входящие в первичную группу создателя-владельца КС» (CREATOR GROUP). Автоматически могут создаваться несколько групп пользователей — «Пользователи» (Users), «Опытные пользователи» (Power Users), «Администраторы» (Administrators), «Операторы архива» (Backup Operators), «Гости» (Guests), «Операторы настройки сети» (Network Operators), «Пользователи удаленного рабочего стола» (Remote Desktop Users), «Репликатор» (Replicator), «Debugger Users» (имеющие право использовать отладчики систем программирования Microsoft), «HelpServices- Group» (группа для центра справки и поддержки).

На серверах КС вместо группы опытных пользователей создаются автоматически группы администраторов учетных записей (Account Operators), администраторов печати (Print Operators) и операторов сервера (Server Operators), а также некоторые другие группы.

На контроллере домена автоматически создаются также группы «Администраторы домена», «Администраторы предприятия», «Администраторы схемы», «Владельцы-создатели групповой политики», «Гости домена», «Издатели сертификатов», «Пользователи домена», «Пользователи журналов производительности», «Пользователи системного монитора» и некоторые другие.

При добавлении в группу учетной записи пользователя или группы (рис. 3.4) возможен выбор размещения (места регистрации учетной записи) — имени локального компьютера или домена, просмотр списка зарегистрированных

учетных записей и выбор среди них нужной (кнопки «Дополнительно» и «Поиск»).

Для глобальной учетной записи с помощью оснастки «Active Directory: пользователи и компьютеры» можно указать дни и часы, в которые разрешен вход в домен под этой учетной записью, а также имена компьютеров, на которых возможен такой вход.

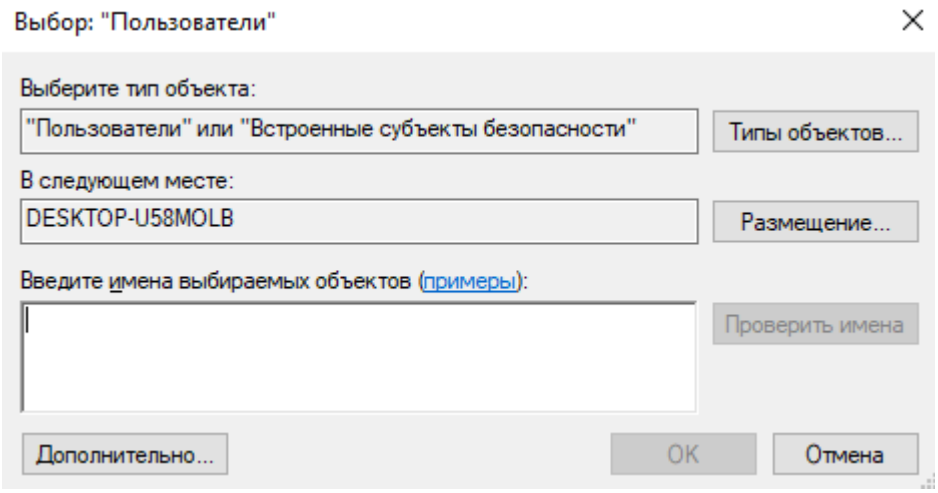

Рисунок 3.4. Добавление учетной записи в группу.

В параметрах локальной политики безопасности (политики безопасности домена) определены две группы параметров учетных записей — параметров политики паролей (рис. 3.5) и параметров политики блокировки учетных записей.

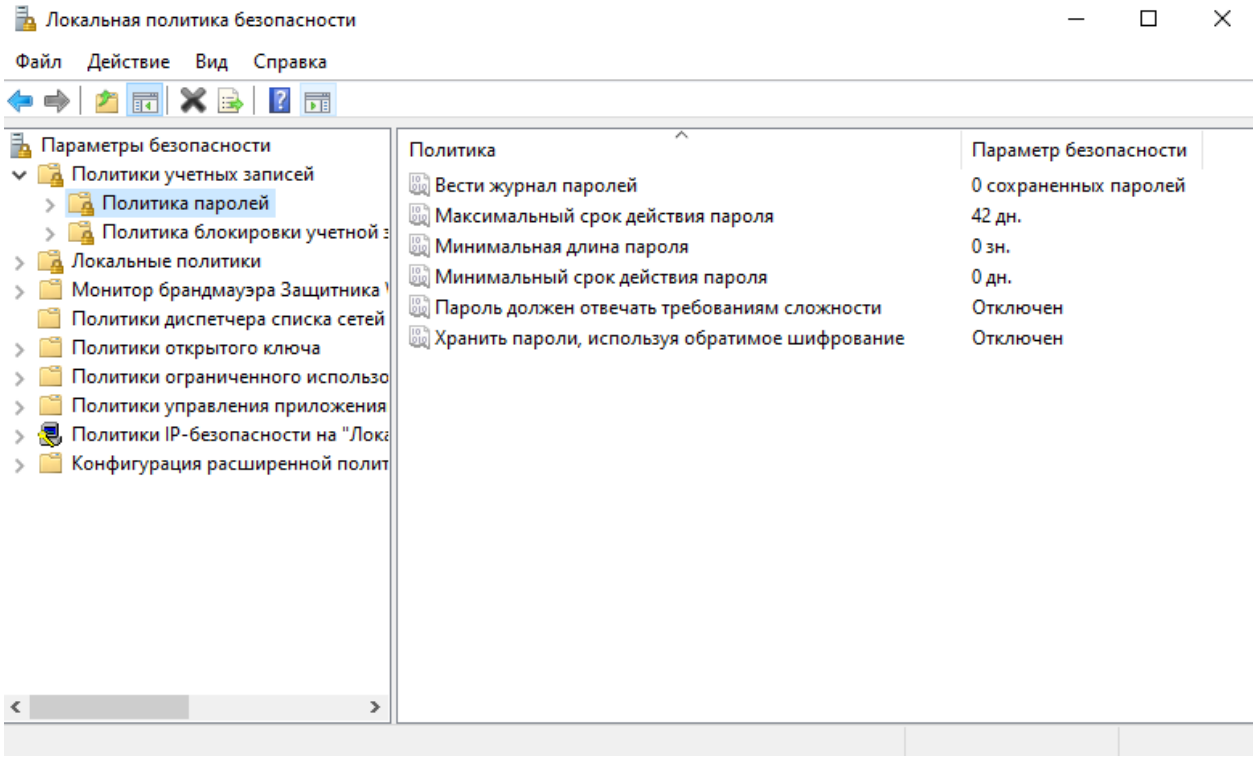

Рисунок 3.5. Параметры политики паролей.

Включение параметра «Пароль должен удовлетворять требованиям сложности» обеспечит выбор пользователями паролей, удовлетворяющих следующим условиям:

– минимальная длина пароля равна шести символам;

– пароль должен включать в себя символы хотя бы трех алфавитов из четырех – строчных латинских букв, прописных латинских букв, цифр и специальных знаков:

– пароль не должен совпадать с логическим именем пользователя или его частью.

С учетом изложенного ранее алгоритма хеширования паролей LAN Manager для повышения безопасности парольной аутентификации требуется дополнительно к данным требованиям сложности паролей установить равной восьми символам их минимальную длину.

По умолчанию на рабочих станциях Windows установлен только параметр, задающий максимальный срок действия пароля (42 дня), а остальные параметры отключены. На контроллере домена для глобальных учетных записей

установлены следующие значения параметров парольной аутентификации: максимальный срок действия пароля — 42 дня, минимальная длина — 7 знаков, минимальный срок действия — 1 день, требовать сложность пароля — включен, требовать неповторяемость паролей — 24 хранимых пароля, хранить пароли с помощью обратимого шифрования — отключен.

К параметрам политики блокировки учетной записи, обеспечивающим защиту от угадывания пароля нарушителем в интерактивном режиме (рис. 1.6), относятся:

– пороговое значение блокировки (максимальное число ошибок входа в систему);

– длительность блокировки учетной записи;

– интервал времени, через который происходит сброс счетчика блокировок.

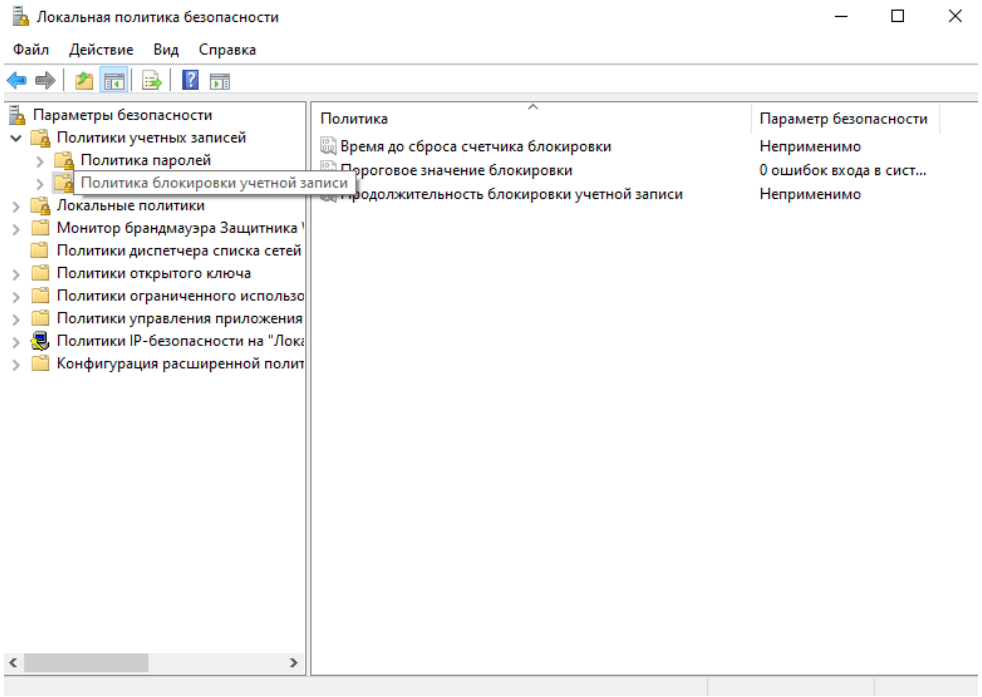

Рисунок 3.6. Параметры политики блокировки учетной записи.

Второй и третий параметры могут быть заданы только после установки ненулевого значения первому параметру (после определения порогового значения блокировки).

С помощью оснастки «Локальная политика безопасности» могут быть определены политики ограниченного использования программ. В этом случае имеется возможность защищать компьютерную систему от программ неизвестного происхождения путем определения программ, разрешенных для запуска. В данной политике приложения могут быть определены с помощью правила для хеша, правила для сертификата, правила для пути и правила для зоны Интернета. Программное обеспечение может выполняться на двух уровнях – неограниченном, при котором доступ программы к ресурсам системы определяется правами пользователя, и «не разрешено», при котором программа запускаться не будет вне зависимости от прав доступа пользователя.

Правила ограниченного использования программ разрешают или запрещают запуск программ, идентифицируемых по контрольному хеш-значению, электронной цифровой подписи определенного субъекта (применяется только к сценариям и пакетам установщика Windows), пути к файлу программы (при перемещении файла правило действовать не будет), зоне безопасности обозревателя Интернета (только для пакетов установщика Windows).

Политика ограниченного использования программ состоит из применяемого по умолчанию правила, используемого при запуске приложений пользователями, и списка (возможно, пустого) программ, являющихся исключением из данного правила. Применяемыми по умолчанию правилами могут быть:

– разрешение на выполнение пользователями всех программ за исключением некоторых;

– запрещение выполнения пользователями всех программ, кроме некоторых.

Список программ, являющихся исключением из применяемого по умолчанию правила, формируется на основе одного из перечисленных выше правил идентификации программ.

Разграничение прав пользователей в операционной системе Windows может быть реализовано двумя способами:

– ограничение прав пользователя или группы на выполнение некоторых функций операционной системы и входящих в ее состав программ (проводника, редактора реестра, обозревателя Интернета и др.). Информация об этих ограничениях сохраняется в профиле пользователя;

– назначение прав пользователю или группе на совершение небезопасных действий в системе. Информация о назначенных правах (привилегиях, privileges) сохраняется в самой учетной записи пользователя или группы.

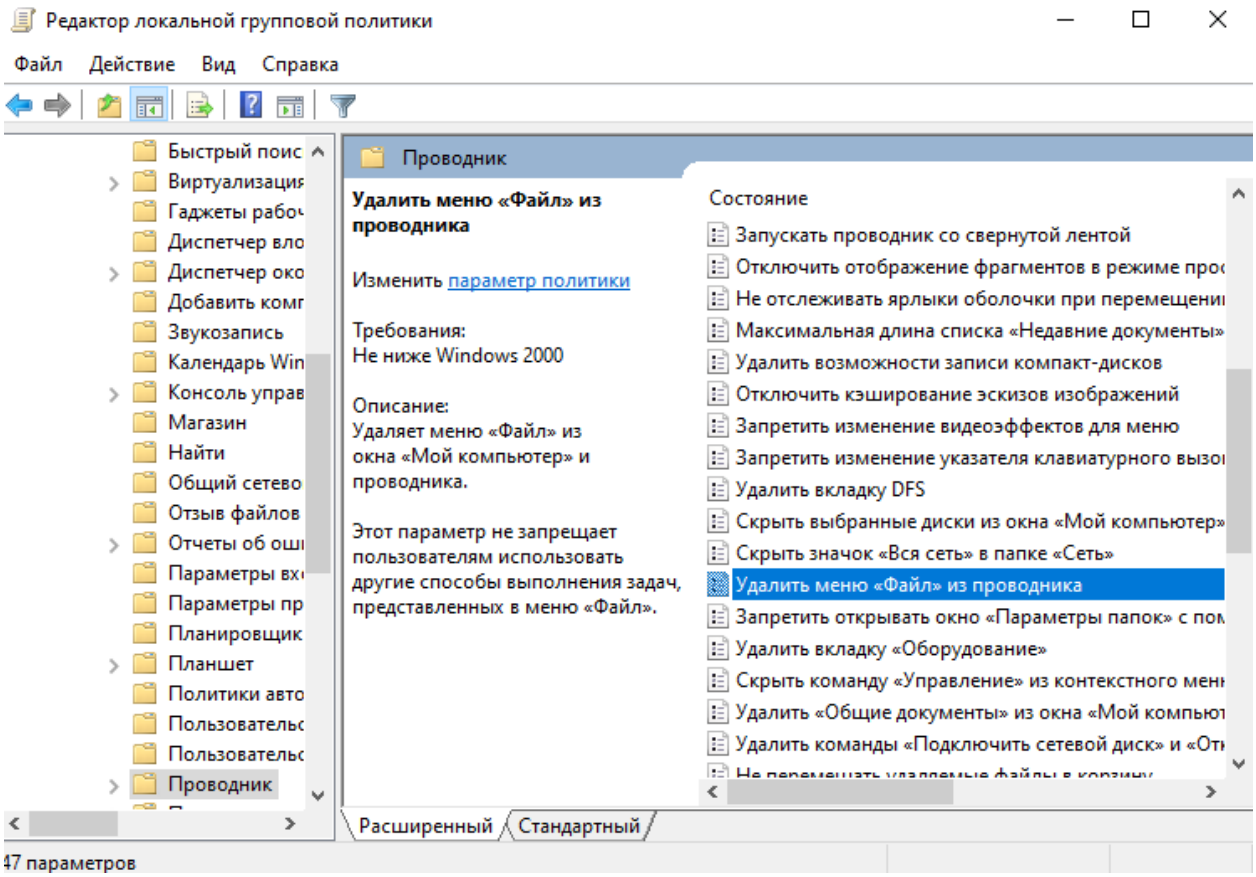

Рисунок 3.7. Ограничение прав пользователя.

Ограничение прав выполняется с помощью оснастки «Редактор объектов групповой политики» (рис. 3.7). Для конкретного пользователя (группы) КС может быть установлен:

– запрет использования средств редактирования реестра;

– запрет использования панели управления или ее отдельных функций;

– удаление команд «Выполнить» и «Поиск» из меню «Пуск»;

– запрет на доступ к отдельным функциям процесса входа (изменению пароля, вызову диспетчера задач, завершению сеанса, блокировке компьютера);

– запрет на настройку панели задач;

– запрет на отображение структуры локальной сети;

– скрытие дисков в папке «Мой компьютер» и в окне проводника;

– скрытие всех объектов на рабочем столе и др.

С помощью оснастки «Редактор объектов групповой политики» также возможны определения перечня доступных для пользователя приложений и механизмов публикации приложений и назначения приложений пользователям и (или) компьютерам. Этот метод ограничения прав пользователя основан на технологии, поддерживаемой установщиком Windows (Windows Installer).

При публикации приложений (используется узел «Конфигурация пользователя») пользователь сам может выбрать программы для установки из списка доступных с помощью функции «Установка и удаление программ» панели управления Windows.

При назначении приложения пользователю в конфигурации пользователя создается перечень обязательных для установки приложений, и при входе пользователя в систему на его рабочий стол помещаются соответствующие ярлыки. Установка приложений производится после щелчка на ярлыке приложения или ассоциированного с ним документа.

Для всех пользователей выбранного компьютера может быть установлен:

– запрет использования перемещаемого профиля пользователя;

– автоматическое обновление системы безопасности и другие важные обновления ОС Windows;

– поддержка «Центра обеспечения безопасности», если компьютер присоединен к домену Active Directory (при включении этой службы происходит наблюдение за основными параметрами безопасности — брандмауэра,

«антивируса», автоматического обновления — и уведомление пользователей, если компьютер подвержен опасности) и др.

При назначении приложения компьютеру (используется узел «Конфигурация компьютера») установка приложений происходит в момент загрузки операционной системы, а удаление установленных программ возможно только с правами локального администратора.

Наложенные ограничения сохраняются в объекте групповой политики. Существует два типа объектов групповой политики. Глобальные объекты групповой политики хранятся на контроллере домена и доступны только в среде Active Directory. Они применяются к пользователям или компьютерам в сайте, домене или подразделении, связанном с объектом групповой политики.

Локальные объекты групповой политики хранятся на локальном компьютере (в разделах реестра HKEY_CURRENT_USER SoftwarePolicies и HKEY_LOCAL_MACHINESoftwarePolicies). На компьютере существует только один локальный объект групповой политики, содержащий набор параметров, доступных в глобальном объекте групповой политики. В случае конфликта параметры локального объекта будут перезаписаны глобальными параметрами или применены совместно.

Назначение прав выполняется с помощью оснастки «Локальная политика безопасности» («Политика безопасности домена») — рис. 3.8. Права (привилегии) пользователя и группы представляют собой права субъектов на выполнение действий, относящихся к системе в целом, а не к отдельным ее объектам. Каждой привилегии соответствует уникальный локальный идентификатор LUID (locally unique identifier), определяемый строкой символов (например, SE SYSTEMTIME NAME). Для отображения пользователю содержания конкретной привилегии с ней также связывается текстовая строка (например, «Изменение системного времени» или «Change the system time»). Внутреннее представление информации о привилегии зависит от конкретной реализации и не документировано Microsoft.

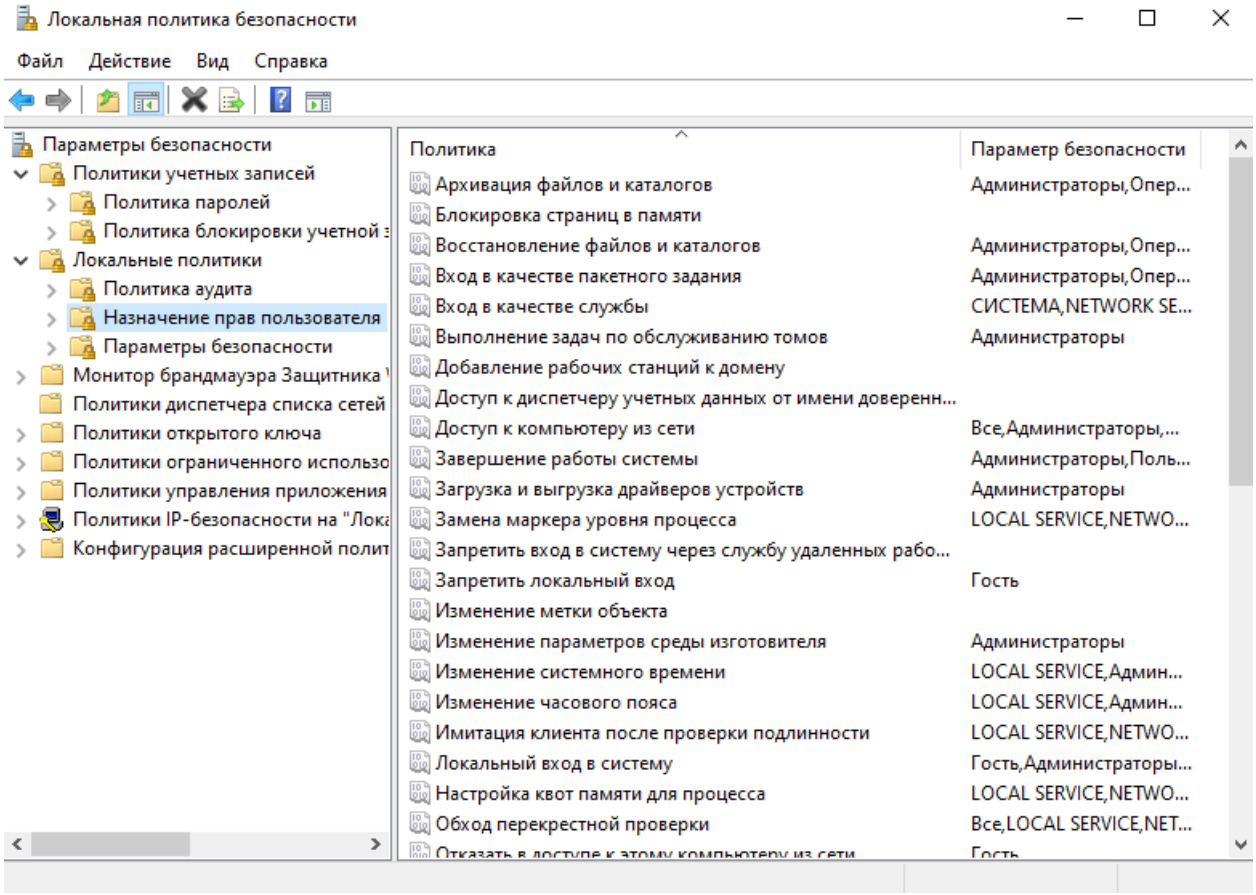

Рисунок 3.8. Назначение прав пользователям и группам.

Перечислим наиболее важные привилегии, которые могут быть назначены пользователям и группам:

– завершение работы системы (SE_SHUTDOWN_NAME);

– изменение системного времени (SE_SYSTEMTIME_NAME);

– отладка программ (SE_DEBUG_NAME) — обладание этой привилегией проверяется только при использовании для отладки систем программирования корпорации Microsoft;

– архивирование файлов и каталогов (SE_BACKUP_NAME);

– восстановление файлов и каталогов (SE_RESTORE_NAME);

– управление аудитом и журналом безопасности (SE_SECURITY_NAME);

– создание журналов безопасности (SE_AUDIT_NAME) — запись в файл аудита информации о событии, связанном с безопасностью системы;

– овладение файлами или иными объектами (SE_TAKE_OWNERSHIP_NAME);

– принудительное удаленное завершение (SE_REMOTE_SHUTDOWN_NAME);

– профилирование загруженности системы (SE_SYSTEM_PROFILE_NAME);

– профилирование одного процесса (SE_PROF_SINGLE_PROCESS_NAME);

– работа в режиме операционной системы (SE_TCB_NAME);

– создание маркерного объекта (SE_CREATE_TOKEN_NAME);

– замена маркера уровня процесса (SE_ASSIGNPRIMARYTOKEN_NAME);

– увеличение приоритета диспетчирования (SE_INC_BASE_PRIORITY_NAME);

– изменение параметров среды оборудования (SE_SYSTEM_ENVIRONMENT_NAME);

– загрузка и выгрузка драйверов устройств (SE_LOAD_DRIVER_NAME);

– вход в качестве пакетного задания (SE_BATCH_LOGON_NAME);

– вход в качестве службы (SE_SERVICE_LOGON_NAME);

– добавление рабочих станций к домену (SE_MACHINE_ACCOUNT_NAME);

– доступ к компьютеру из сети (SE_NETWORK_LOGON_NAME);

– закрепление страниц в памяти (SE_LOCK_MEMORY_NAME);

– запретить вход в систему через службу терминалов (SE_DENY_REMOTE_INTERACTIV_LOGON_NAME);

– запуск операций по обслуживанию тома (SE_MANAGE_VOLUME_NAME);

– локальный вход в систему (SE_INTERACTIVE_LOGON_NAME);

– отказ в доступе к компьютеру из сети (SE_DENY_NETWORK_LOGON_NAME);

– отказ во входе в качестве пакетного задания (SE_DENY_BATCH_LOGON_NAME);

– отказать во входе в качестве службы (SE_DENY_SERVICE_LOGON_NAME);

– отклонить локальный вход (SE_DENY_INTERACTIVE_LOGON_NAME);

– разрешать вход в систему через службу терминалов (SE_REMOTE_INTERACTIVE_LOGON_NAME);

– разрешение доверия к учетным записям при делегировании (SE_ENABLE_DELEGATION_NAME);

– синхронизация данных службы каталогов (SE_SYNC_AGENT_NAME);

– создание глобальных объектов (SE_CREATE_GLOBAL_NAME);

– создание постоянных объектов совместного использования (SE_CREATE_PERMANENT_NAME);

– создание страничного файла (SE_CREATE_PAGEFILE_NAME).

С помощью механизма назначения прав на вход в систему можно обеспечить дополнительную защиту отдельных компьютеров сети, а также уменьшить риск, связанный с использованием административных привилегий. Например, право SE_NETWORKLOGONNAME может быть назначено всем учетным записям, а привилегия SE_DENY_NETWORK_LOGON_ NAME — членам группы администраторов, чтобы запретить удаленное администрирование данного компьютера.

Обладание правами, связанными с использованием наиболее важных для безопасности функций системы, должно быть ограничено, поскольку позволяет субъектам обходить те или иные средства защиты информации в КС. Например, привилегии архивирования и восстановления файлов и каталогов позволяют их обладателям обходить правила разграничения доступа к объектам КС. Привилегия увеличения приоритета диспетчирования позволяет ее обладателю

нарушить доступность информации в операционной системе, создав процесс с максимально высоким приоритетом и организовав в нем бесконечный цикл. Привилегия создания журналов безопасности позволяет ее обладателю записывать в файл аудита информацию для компрометации других пользователей и скрытия собственных действий и т. д.

3.2. Разграничение доступа к объектам в операционной системе Windows

Для разграничения доступа субъектов к объектам КС в защищенных версиях операционной системы Windows используется дискреционное управление доступом. С каждым объектом разграничения доступа связывается дескриптор безопасности SD (security descriptor), содержащий следующую информацию:

– идентификатор безопасности (SID) владельца объекта;

– идентификатор безопасности первичной группы владельца (в Windows это поле не используется в алгоритме проверки прав доступа субъекта к объекту);

– дискреционный список контроля доступа (discretionary access control list – DACL);

– системный список контроля доступа (system access control list – SACL).

Список SACL управляется администратором системы. Список DACL предназначен для идентификации пользователей и групп, которым предоставлен или запрещен определенный тип доступа к объекту. Этот список редактируется владельцем объекта, но и члены группы администраторов по умолчанию имеют право на смену разрешений на доступ к любому объекту, которое может быть у них отнято владельцем объекта.

Каждый элемент списка DACL (access control entry — ACE) определяет права доступа к объекту одного пользователя или группы. Каждый ACE содержит следующую информацию:

– идентификатор безопасности SID субъекта, для которого определяются права доступа;

– маску доступа (access mask — AM), которая специфицирует контролируемые данным ACE права доступа;

– тип ACE;

– признак наследования прав доступа к объекту, определенных для родительского объекта.

Элементы списка DACL могут быть двух типов – элементы, разрешающие специфицированные в них права доступа (Access-allowed ACE), и элементы, запрещающие определенные в них права доступа (Access-denied ACE). Элементы для запрещения субъектам использования определенных прав доступа должны размещаться в «голове» списка, до первого из элементов, разрешающих использование субъектом тех или иных прав доступа.

Право доступа субъекта к объекту означает возможность обращения субъекта к объекту с помощью определенного метода (типа) доступа. В операционной системе Windows различаются специальные, стандартные (общие) и родовые (generic) права доступа к объектам. Специальные права доступа определяют возможность обращения к объекту по свойственному только данной категории объектов методу – чтение данных из объекта, запись данных в объект, чтение атрибутов объекта, выполнение программного файла и т. д. Стандартные права доступа определяют возможность доступа к объекту по методу, применимому к любому объекту, – изменение владельца объекта, изменение списка DACL объекта, удаление объекта и т. д.

Каждое из родовых прав доступа представляет собой комбинацию специальных и стандартных прав и предоставляет возможность обращения к объекту с помощью некоторого набора методов доступа. Определены следующие родовые права доступа:

1. Для файлов и папок:

– полный доступ (включает в себя все специальные и стандартные разрешения);

– изменение (все разрешения, кроме «Удаление подпапок и файлов», «Смена разрешений» и «Смена владельца»);

– чтение и выполнение (включает разрешения на «Обзор папок (выполнение файлов)», «Содержание папки (чтение данных)», «Чтение атрибутов», «Чтение дополнительных атрибутов», «Чтение разрешений», «Синхронизация»);

– список содержимого папки (только для папок); включает в себя те же разрешения, что и «Чтение и выполнение», но они наследуются по-разному. Разрешение «Список содержимого папки» наследуется только папками, но не файлами, и отображается только при просмотре разрешений на доступ к папкам. Разрешение «Чтение и выполнение» наследуется как файлами, так и папками, и всегда отображается при просмотре разрешений на доступ к файлам или папкам;

– чтение (включает в себя право на «Содержание папки (чтение данных)», «Чтение атрибутов», «Чтение дополнительных атрибутов», «Чтение разрешений», «Синхронизацию»);

– запись (включает в себя разрешения «Создание файлов (запись данных)», «Создание папок (дозапись данных)», «Запись атрибутов», «Запись дополнительных атрибутов», «Чтение разрешений», «Синхронизацию»).

2. Для принтеров (родовые права доступа):

– печать (включает разрешения на «Печать» и «Чтение разрешений»);

– управление документами (включает в себя права на «Управление документами», «Чтение разрешений», «Смену разрешений», «Смену владельца»);

– управление принтерами (включает в себя все специальные и стандартные права доступа, кроме «Управления документами»).

3. Для реестра Windows (родовые права доступа):

– полный доступ (включает в себя все специальные и стандартные разрешения);

– чтение (включает разрешения на «Запрос значения», «Перечисление подразделов», «Уведомление», «Чтение разрешений»).

Специальное право на «Обзор папок» предполагает возможность перемещения по структуре папок в поисках других файлов или папок, даже если пользователь не обладает разрешением на доступ к просматриваемым папкам. Право на «Содержание папки» дает возможность просмотра имен файлов и подпапок, содержащихся в папке. Это разрешение относится только к содержимому данной папки и не означает, что имя самой этой папки также должно включаться в список.

Специальное право «Выполнение файлов» разрешает или запрещает запуск программ. Разрешение «Обзор папок» для папки не означает автоматическую установку разрешения «Выполнение файлов» для всех файлов в этой папке. Право на «Запись данных» разрешает или запрещает внесение изменений в файл и запись поверх имеющегося содержимого. Право на «Дозапись данных» разрешает или запрещает внесение данных в конец файла, но не изменение, удаление или замену имеющихся данных.

Назначение дескрипторов безопасности вновь создаваемым объектам в защищенных версиях операционной системы Windows производится по следующим правилам:

– на основе явно заданного субъектом и корректного по форме дескриптора безопасности (например, при вызове системных функций CreateFile или CreateDirectory при создании файлов или папок, при вызове системной функции RegCreateKeyEx при создании раздела реестра и т. п.);

– на основе механизма наследования (если при создании объекта дескриптор безопасности не задается);

– из маркера доступа субъекта, создающего объект (если наследование невозможно).

Рассмотрим более подробно механизм наследования прав доступа из дескриптора безопасности родительского объекта. Наследование дескриптором безопасности дочернего объекта элементов ACE из списка DACL дескриптора безопасности объекта-родителя зависит от следующих флагов, содержащихся в заголовке ACE:

– CONTAINER_INHERIT_ACE – данный ACE наследуется контейнерным дочерним объектом;

– OBJECT_INHERIT_ACE – данный ACE наследуется неконтейнерным дочерним объектом;

– INHERIT_ONLY_ACE – данный ACE не используется при проверке прав доступа к родительскому объекту и используется только при наследовании разрешений дочерними объектами;

– NO_PROPAGATE_INHERIT_ACE – флаги OBJECT_INHERIT_ACE и CONTAINER_INHERIT_ACE сбрасываются при наследовании данного ACE.

Если в маске доступа ACE родительского объекта содержатся родовые права доступа, то перед наследованием данного ACE происходит их отображение.

В Windows возможно разграничение доступа к следующим объектам:

– файлам и папкам на дисках под управлением файловой системы NTFS;

– разделам реестра;

– принтерам;

– трубам (pipes);

– процессам и потокам (нитям, threads);

– файлам, отображенным на память (file-mapping objects);

– маркерам доступа;

– службам (services);

– объектам состояния окна (window station objects) – буферу обмена, глобальным атомам, объектам рабочего стола;

– рабочим столам;

– объектам синхронизации (событиям, мьютексам, семафорам);

– объектам задач.

Пользователи КС для назначения субъектам КС прав доступа к файлам и папкам на дисках с файловой системой NTFS, чьими создателями/владельцами они являются, должны применять средства Проводника Windows. Для этого

(рис. 3.9) выполняются команды «Свойства» контекстного меню папки либо команда «Свойства» контекстного меню файла, вкладка «Безопасность».

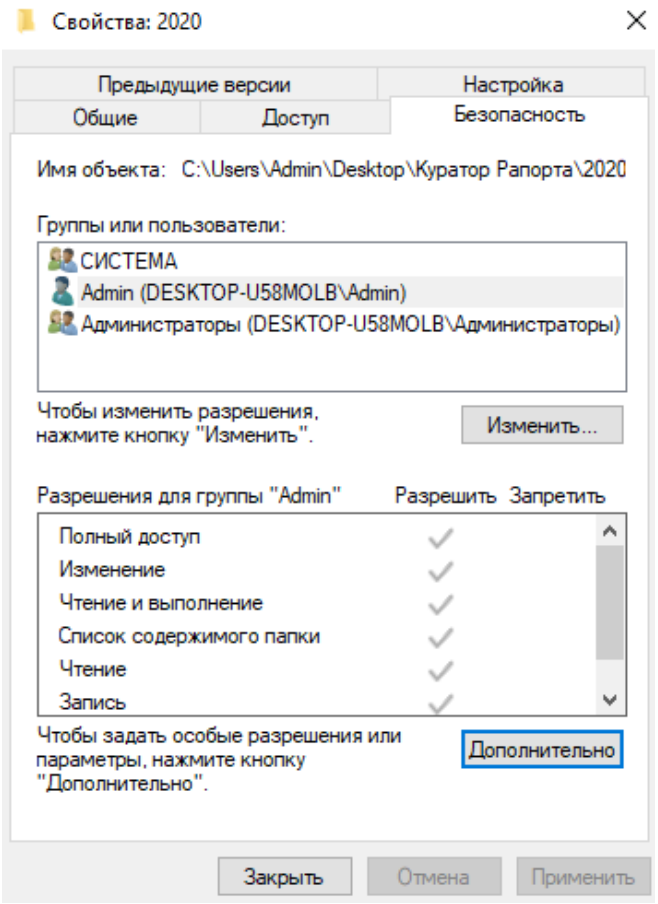

Рисунок 3.9. Управление разрешениями.

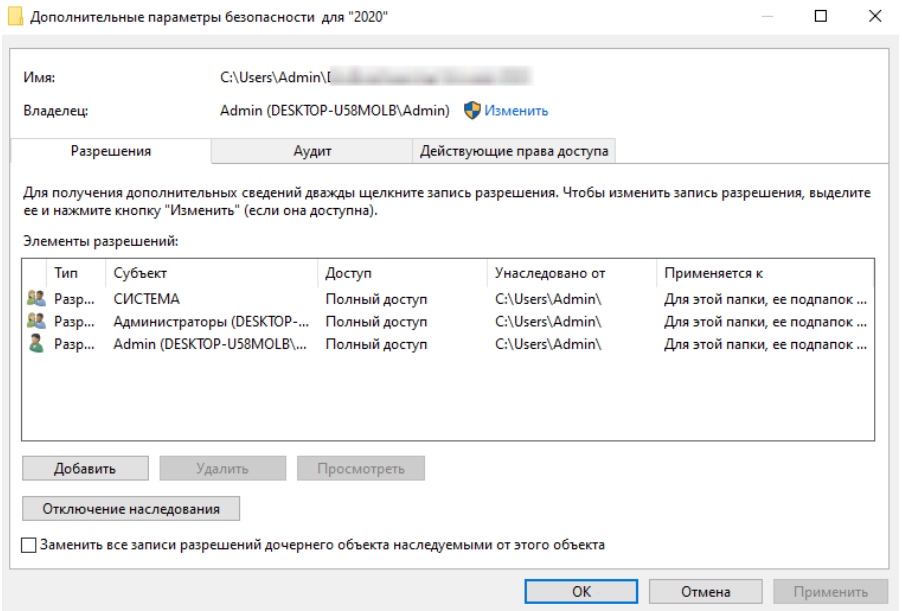

Рисунок 3.10. Просмотр установленных разрешений.

Кнопки «Добавить» и «Удалить» позволяют изменять количество элементов ACE в списке DACL, а в окне «Разрешения для ...» можно устанавливать родовые права доступа к объекту конкретным пользователям и группам.

На вкладке «Действующие разрешения» (рис. 3.11) можно проверить, какие права доступа к объекту установлены для конкретного пользователя или группы, выбирающиеся с помощью кнопки «Выбрать». Действующие разрешения определяются с учетом членства субъекта доступа в различных группах.

Установка прав доступа субъектов КС к установленным в системе принтерам, системным службам и разделам реестра производится аналогично с помощью функций панели управления и редактора реестра (regedit.exe).

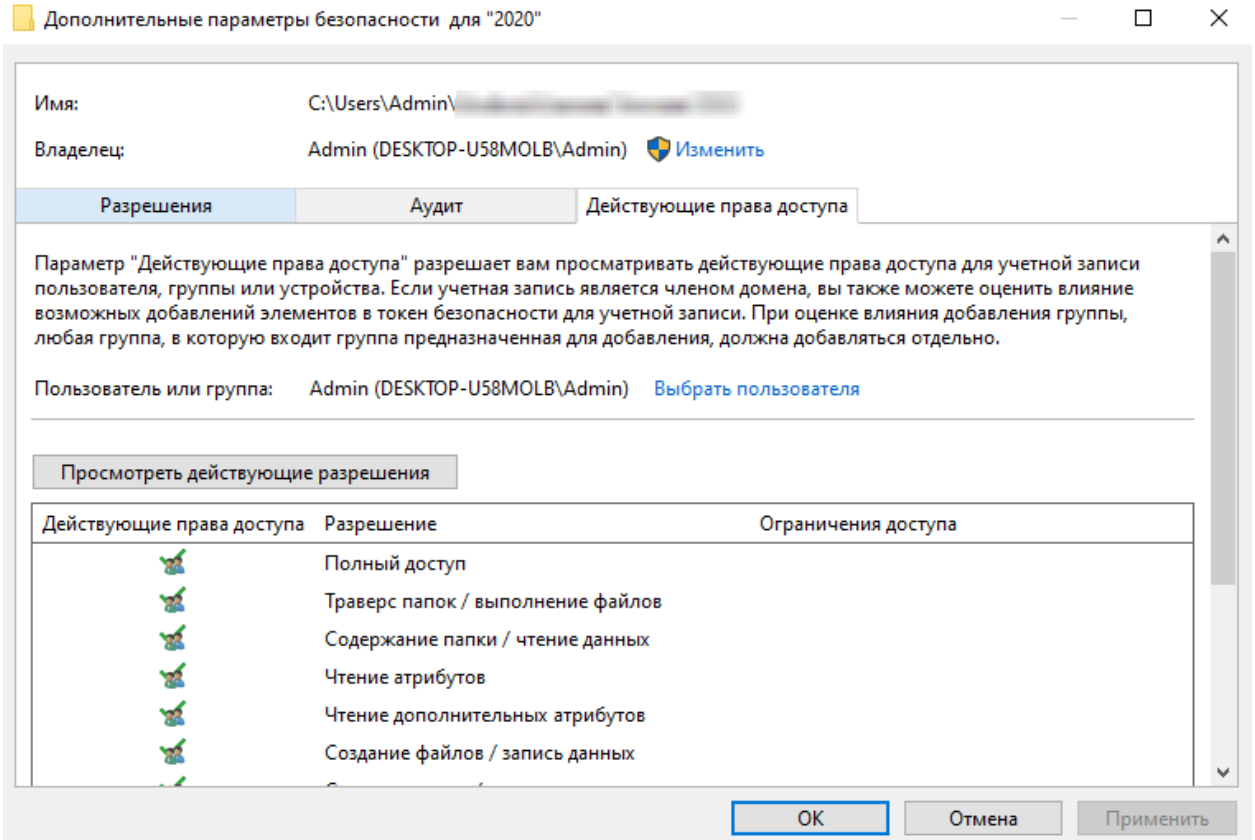

Рисунок 3.11. – Просмотр разрешений для выбранного субъекта.

Разграничение доступа к разделам реестра возможно при любой файловой системе, используемой на диске с реестром.

Проводник и редактор системного реестра Windows правильно отображают список DACL объекта только при выполнении двух условий:

– все элементы ACE, запрещающие доступ к объекту, находятся в «голове» списка DACL;

– в элементах ACE, запрещающих доступ субъектов к объекту, запрещены все права доступа, а не избранные.

К недостаткам разграничения доступа к объектам КС, реализованного в защищенных версиях операционной системы Windows, можно отнести то, что пользователи КС не имеют средств управления правами доступа субъектов к объектам других типов (процессам и потокам, рабочим столам и оконным объектам, маркерам доступа, объектам синхронизации и др.).

3.3. Аудит событий безопасности в операционной системе Windows

Важным элементом администрирования элементов обеспечения информационной безопасности является аудит событий в системе. Настройка данного параметра осуществляется с помощью консоли локальной политики безопасности. Чтобы открыть локальную политику безопасности необходимо в командной строке ввести ***secpol.msc*** и нажать клавишу «ввод» (рис. 3.12). Также настройка параметров политики безопасности может быть осуществлена с помощью консоли редактора локальной групповой политики. Чтобы открыть редактор локальных политик безопасности необходимо в командной строке ввести ***gpedit.msc*** и нажать клавишу «ввод». В дереве консоли «Конфигурация компьютера», выбрать пункт «Конфигурация Windows», а затем – «Параметры безопасности» (рис 3.13). Для изменений политики аудита необходимо выбрать пункт «Локальные политики», а затем – «Политика аудита».

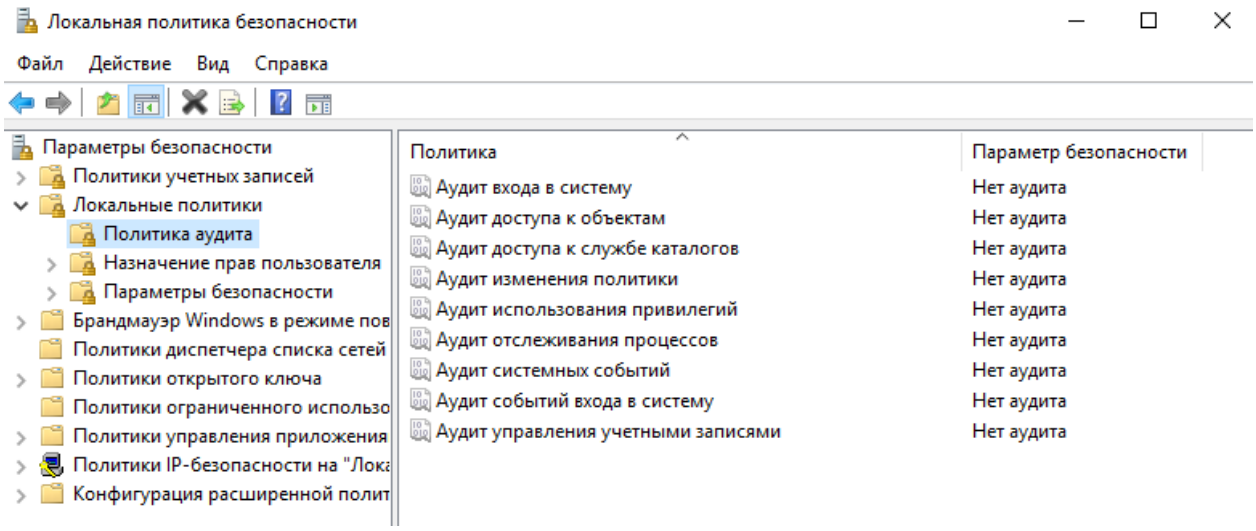

Рисунок 3.12. Настройка политики аудита событий в системе с помощью консоли локальной политики безопасности.

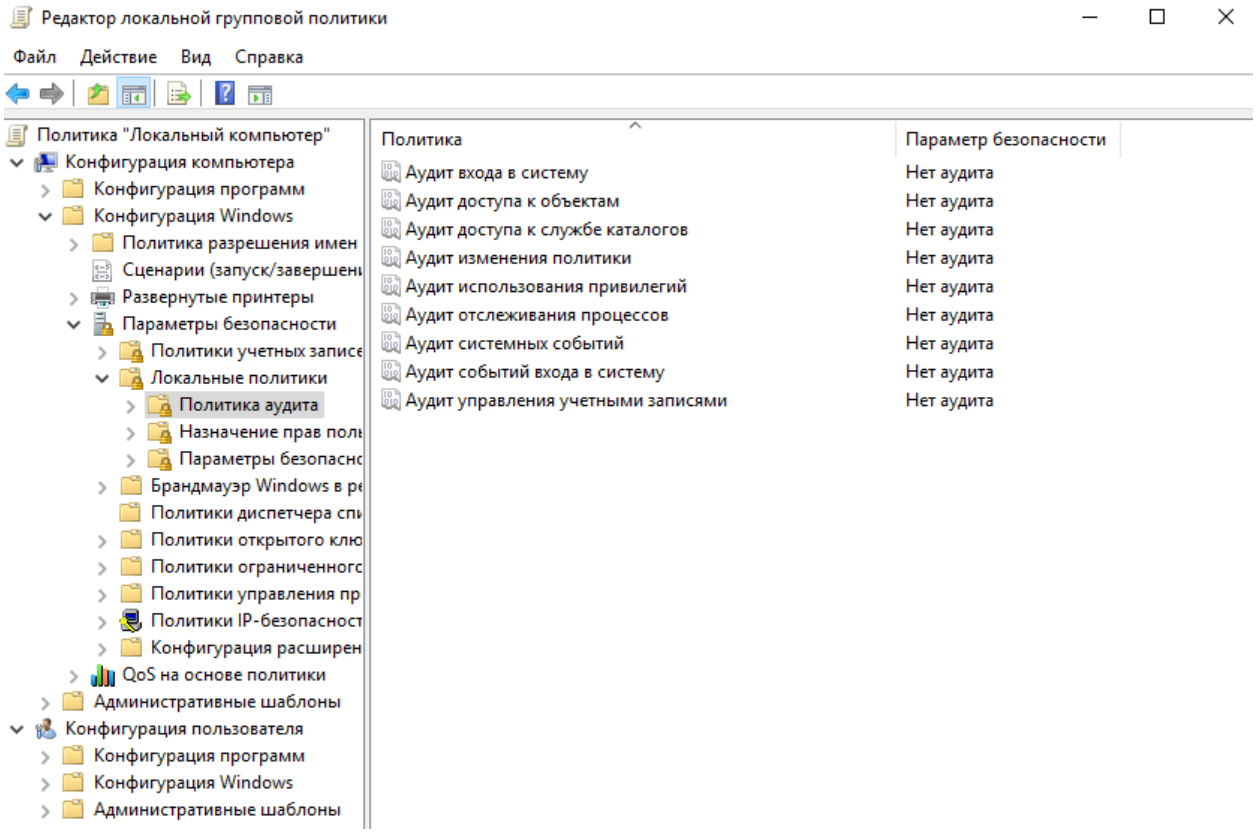

Рисунок 3.13. Настройка политики аудита событий в системе с помощью консоли редактора локальной групповой политики.

В операционной системе Windows ведется аудит событий по следующим категориям:

- аудит входа в систему;
- аудит доступа к объектам;
- аудит доступа к службе каталогов;
- аудит изменения политики;
- аудит использования привилегий;
- аудит отслеживания процессов;
- аудит системных событий;
- аудит входа в систему;
- аудит управления учетными записями.

Рассмотрим назначение каждого параметра подробнее.

3.3.1. Аудит входа в систему

Параметр определяет, будет ли операционная система выполнять аудит каждой попытки входа пользователя в систему или выхода из нее на данном компьютере.

События выхода из системы создаются каждый раз, когда завершается сеанс вошедшей в систему учетной записи пользователя. Если этот параметр политики задан, то администратор может указать, какие события подвергаются аудиту: только об успешном выполнении, только об ошибках, те и другие (также можно отключить аудит данного типа событий).

Значения по умолчанию для клиентских выпусков:

- Вход в систему: аудит событий об успешном выполнении.
- Выход из системы: аудит событий об успешном выполнении.
- Блокировка учетных записей: аудит событий об успешном выполнении.
- Основной режим IPsec: аудит не выполняется.
- Быстрый режим IPsec: аудит не выполняется.

– Расширенный режим IPsec: аудит не выполняется.

– Специальный вход: аудит событий об успешном выполнении.

– Другие события входа и выхода: аудит не выполняется.

– Сервер политики сети: аудит как событий об успешном выполнении, так и об ошибках.

3.3.2. Аудит доступа к объектам

Этот параметр безопасности определяет, будет ли операционная система выполнять аудит попыток доступа пользователей к объектам, не относящимся к Active Directory. Аудит создается только для объектов, для которых указаны списки управления доступом, при условии, что запрашиваемый тип доступа (например, «Запись», «Чтение» и «Изменение») и учетная запись, выполняющая запрос, соответствуют параметрам в данных списках.

Администратор может задать только аудит успехов, только аудит неудач либо отключить аудит этих событий совсем (как для успехов, так и неудач).

Если включен аудит успехов, запись аудита создается при каждой успешной попытке доступа учетной записи к объекту, не относящемуся к Active Directory, соответствующему заданному списку управления доступом.

Если включен аудит неудач, запись аудита создается при каждой неудачной попытке доступа учетной записи к объекту, не относящемуся к Active Directory, соответствующему заданному списку управления доступом.

Обратите внимание, что можно задать список управления доступом для объекта файловой системы с помощью вкладки "Безопасность" в диалоговом окне свойств этого объекта.

Значение по умолчанию: нет аудита.

3.3.3. Аудит доступа к службе каталогов

Этот параметр безопасности определяет, будет ли операционная система выполнять аудит попыток доступа пользователей к объектам Active Directory. Аудит создается только для объектов, для которых указан системный список управления доступом, при условии, что запрашиваемый тип доступа (например, «Запись», «Чтение» или «Изменение») и учетная запись, выполняющая запрос, соответствуют параметрам в данном списке.

Администратор может указать, какие события подвергаются аудиту: только об успешном выполнении, только об ошибках, те и другие (также можно отключить аудит обоих типов данных событий).

Если включен аудит событий об успешном выполнении, то запись аудита создается при каждой успешной попытке доступа учетной записи к объекту Active Directory, соответствующей заданному системному списку управления доступом.

Если включен аудит событий об ошибках, то запись аудита создается при каждой неудачной попытке доступа учетной записи к объекту Active Directory, соответствующей заданному системному списку управления доступом.

Значения по умолчанию для клиентских выпусков:

- Доступ к службе каталогов: аудит не выполняется.
- Изменения службы каталогов: аудит не выполняется.
- Репликация службы каталогов: аудит не выполняется.
- Подробная репликация службы каталогов: аудит не выполняется.

3.3.4. Аудит изменения политики

Этот параметр безопасности определяет, будет ли операционная система выполнять аудит каждой попытки изменения политики назначения прав пользователям, аудита, учетной записи или доверия.

Администратор может задать только аудит успехов, только аудит неудач либо отключить аудит этих событий совсем (как для успехов, так и неудач).

Если включен аудит успехов, запись аудита создается при успешной попытке изменения политики назначения прав пользователям, аудита, учетной записи или доверия.

Если включен аудит неудач, запись аудита создается при попытке изменения политики назначения прав пользователям, аудита, учетной записи или доверия учетной записью, не уполномоченной выполнять запрос изменения политики.

Значения по умолчанию:

- Аудит изменения политики: успех.
- Изменение политики проверки подлинности: успех.
- Изменение политики авторизации: нет аудита.
- Изменение политики на уровне правил MPSSVC: нет аудита.
- Изменение политики платформы фильтрации: нет аудита.
- Другие события изменения политики: нет аудита.

3.3.5. Аудит использования привилегий

Этот параметр безопасности определяет, будет ли выполняться аудит каждого пользователя с помощью применения прав пользователя.

Определяя этот параметр политики, можно задать аудит успехов, аудит неудач либо отключить аудит всех типов событий. Аудиты успехов создают запись аудита в случае успешного применения прав пользователя. Аудиты неудач создают запись аудита в случае неудачного применения прав пользователя.

Чтобы отключить аудит, в диалоговом окне свойств этого параметра политики установите флажок «Задать эти параметры политики» и снимите флажки «Успех» и «Неудача».

Значение по умолчанию: без аудита.

Записи аудита не создаются при использовании следующих прав пользователя, даже если для аудита использования привилегий задан аудит успехов или неудач. Включение аудита этих прав пользователя приведет к записи большого количества событий в журнал безопасности, что может снизить производительность компьютера.

Для аудита следующих прав пользователя включите раздел реестра FullPrivilegeAuditing:

- Обход перекрестной проверки.
- Отладка программ.
- Создание объекта типа токен.
- Замена токена уровня процесса.
- Создание аудитов безопасности.
- Архивация файлов и каталогов.
- Восстановление файлов и каталогов.

Стоит отметить, что неправильное редактирование реестра может серьезно повредить систему. Перед изменением реестра необходимо сохранить все важные данные на компьютере.

3.3.6. Аудит отслеживания процессов

Этот параметр безопасности определяет, будет ли операционная система выполнять аудит событий, связанных с процессами, такими как создание процесса, завершение, обработка дублирований, а также непрямой доступ к объектам.

Если этот параметр политики определен, администратор может задать аудит только успехов, только неудач, успехов и неудач либо отключить аудит этих событий совсем.

Если включен аудит успехов, запись аудита создается при каждом успешном отслеживании операционной системой действий, связанных с процессами.

Если включен аудит неудач, запись аудита создается каждый раз, когда операционная система не может выполнить одно из этих действий.

Значение по умолчанию: нет аудита.

3.3.7. Аудит системных событий

Этот параметр безопасности определяет, будет ли операционная система выполнять аудит следующих событий:

– Попытка изменения системного времени.

– Попытка запуска или отключения системы безопасности.

– Попытка загрузки компонентов расширяемой проверки подлинности.

– Потеря отслеженных событий из-за сбоя системы аудита.

– Размер журнала безопасности превысил настраиваемый уровень порогового значения предупреждений.

Если определен этот параметр политики, администратор может задать аудит только успехов, только неудач, успехов и неудач либо отключить аудит всех этих событий совсем (и успехов, и неудач).

Если включен аудит успеха, создается запись аудита при каждом успешном выполнении операционной системой этих действий.

Если включен аудит неудачи, создается запись аудита при каждой неудачной попытке выполнения операционной системой этих действий.

Значения по умолчанию:

– Изменение состояния безопасности: успех.

– Расширение системы безопасности: нет аудита.

– Целостность системы: успех, неудача.

– Драйвер IPsec: нет аудита.

– Другие системные события: успех, неудача.

3.3.8. Аудит событий входа в систему

Этот параметр безопасности определяет, будет ли операционная система выполнять аудит каждый раз при проверке данным компьютером учетных данных.

События входа учетной записи формируются при проверке доверенным компьютером подлинности учетной записи. Члены домена и компьютеры, не входящие в домен, являются доверенными для своих локальных учетных записей; контроллеры доменов являются доверенными для учетных записей в домене. Проверка учетных данных может быть подтверждением локального входа или, в случае учетной записи в домене Active Directory на контроллере домена, подтверждением входа на другом компьютере. Проверка учетных данных не отслеживается, поэтому для событий входа в систему нет соответствующего события выхода.

Если данный параметр политики задан, то администратор может определить, какие события следует подвергать аудиту: об успешном выполнении, об ошибках, те и другие или никакие (т.е. не производить аудит данных событий).

Значения по умолчанию для клиентских выпусков:

- Проверка учетных данных: аудит не выполняется.
- Операции с билетами службы Kerberos: аудит не выполняется.
- Другие события входа в систему учетной записи: аудит не выполняется.
- Служба проверки подлинности Kerberos: аудит не выполняется.

3.3.9. Аудит управления учетными записями

Этот параметр безопасности определяет, необходимо ли выполнять аудит каждого события управления учетными записями на компьютере. Примеры событий управления учетными записями:

Учетная запись пользователя либо группа создана, изменена или удалена.

Учетная запись пользователя переименована, отключена или включена.

Пароль установлен или изменен.

При определении данного параметра политики можно указать, какие события следует подвергать аудиту: об успешном выполнении или об ошибках (также можно отключить аудит для данного типа событий). Аудит событий об успешном выполнении создает запись аудита, когда любое событие управления учетными записями завершается успешно. Аудит событий об ошибках создает запись аудита, когда любое событие управления учетными записями завершается с ошибкой. Чтобы отключить аудит, в диалоговом окне свойств этого параметра политики выберите флажок «Задать эти параметры политики» и снимите флажки «Успех» и «Неудача».

Значения по умолчанию для клиентских выпусков:

– Управление учетными записями: аудит событий об успешном выполнении.

– Управление учетными записями компьютеров: аудит не выполняется.

– Управление группой безопасности: аудит событий об успешном выполнении.

– Управление группой распространения: аудит не выполняется.

– Управление группой приложений: аудит не выполняется.

– Другие события управления учетными записями: аудит не выполняется.

3.3.10. События аудита

События аудита записываются в журналы следующих типов:

- журнал приложений;
- журнал безопасности;
- журнал системы;
- службы каталогов;
- журнал перенаправленных событий.

Просмотр журнала безопасности осуществляется в оснастке «Просмотр событий» посредством ввода в командной строке команды ***eventvwr.msc*** (рис. 3.14). Сами журналы хранятся в папке %WinDir%\System32\winevt\Logs.

В журнал записываются события трех основных видов6 информационные сообщения о событиях, предупреждающие сообщения о событиях и сообщения о событиях ошибок.

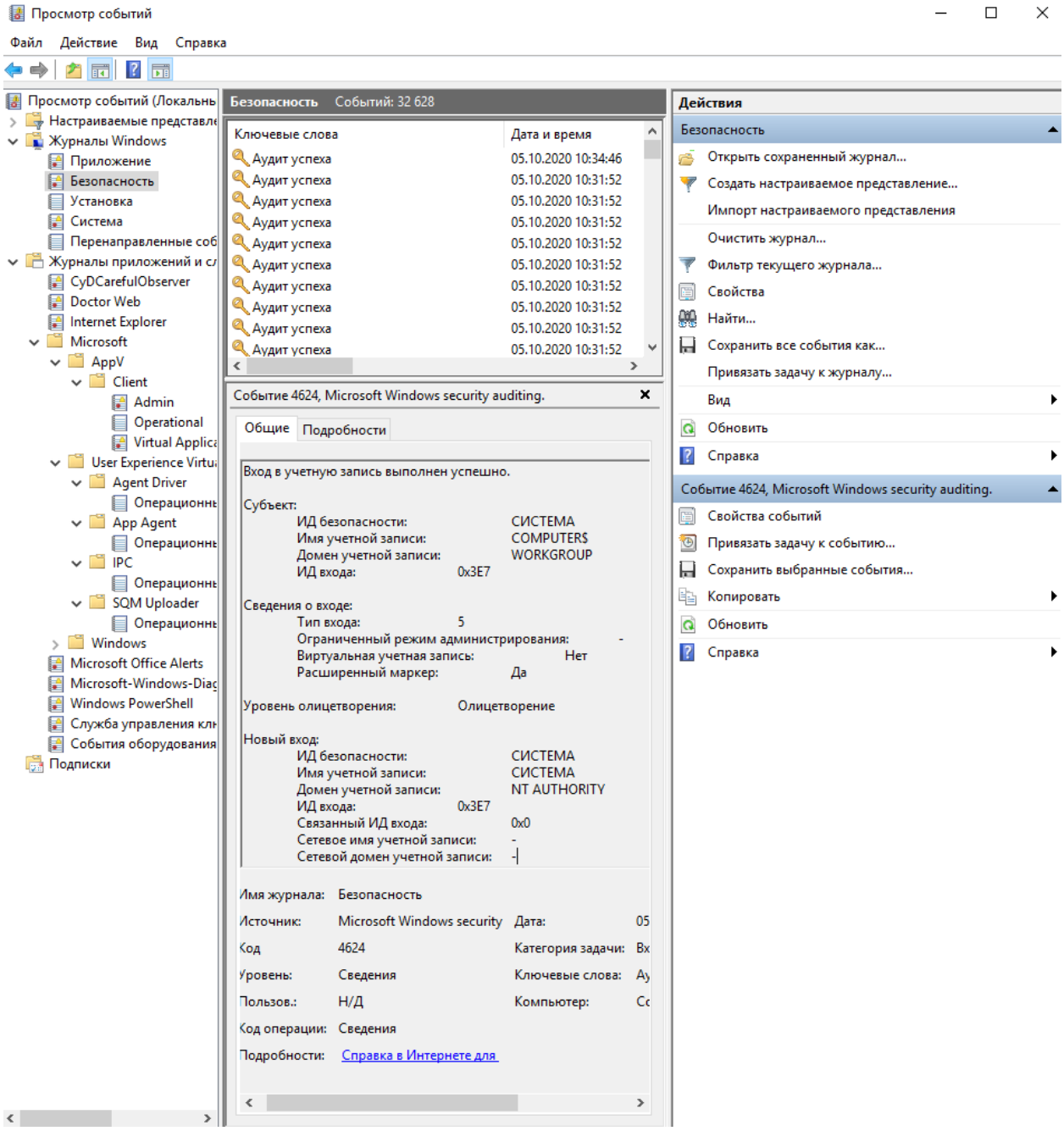

Рисунок 3.14. Оснастка Windows «Просмотр событий».

4. АДМИНИСТРИРОВАНИЕ ЭЛЕМЕНТОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ РАБОЧИХ СТАНЦИЙ НА ОСНОВЕ ОПЕРАЦИОННЫХ СИСТЕМ LINUX

4.1. Особенности операционной системы ОС Linux

4.1.1. Файловая система ОС Linux

Файловая система Linux, в отличие от операционных систем семейства Windows не разделена по томам (дискам, устройствам), а имеет единую древовидную структуру, в основе которой лежит корневой каталог. Корневой каталог — это уровень файловой системы, выше которого по дереву каталогов подняться невозможно. В Linux корневой каталог обозначается как / (именно / - слэш, а не \ - обратный слэш). Система позволяет устанавливать много корневых каталогов. Так, например, для некоторого пользователя ftp /home будет корневым каталогом и при обращении к клиенту ftp на смену каталога на корневой пользователь будет попадать в /home.

Для подключения любого устройства к файловой системе используется так называемая точка монтирования – каталог, все вложенные уровни которого являются файловой системой на устройстве-носителе. Например, при монтировании дискеты обычно используется каталог /media/floppy . То есть, все каталоги и файлы, находящиеся внутри /media/floppy на самом деле содержатся на дискете, вставленной в дисковод компьютера. Для подключения, или монтирования, устройств используется специальная команда, которую мы изучим на следующих занятиях. Таким образом подключаются и сетевые файловые системы, то есть такие системы, которые реально находятся где-то на сервере в сети, однако различий в работе с ними пользователь не ощущает и видит сетевые файлы и каталоги, как если бы они были расположены на локальном компьютере.

Есть у файловой системы Linux и еще одна особенность: пользователям в ней выделяется домашний каталог – специальный каталог, необходимый для

хранения пользователем своих личных данных. При входе пользователя в систему, он сразу оказывается в своем домашнем каталоге. Обычно права доступа к домашнему каталогу с консоли пользователя выставлены таким образом, что доступ к каталогу запрещен всем кроме владельца и администратора.

4.1.2. Типы файлов и устройства в ОС Linux

В файловой системе Линукс различают несколько типов файлов. Понятие "файл" включает в себя также и интерфейсы работы с периферийными устройствами, и каналы, позволяющие разным процессам в системе обмениваться данными.

```
[student@ns lesson_2]$ ls -l
total 40
-rwxr-xr-x    1 root     root         2872 Aug 27  2001 arch
-rw-rw-rw-    1 root     root          612 Jun 25  2001 chain.b
brw-rw----    1 root     disk       3,   1 Feb  3 15:38 hda1
drwxrwxrwx    2 root     root        32768 Feb  3 15:38 ida
```

Листинг 4.1. Перечень устройств ОС Linux.

Таблица 4.1. Основные типы устройств.

Основные типы устройств	
-	простой файл
d	каталог
l	ссылка
b	блочное устройство
c	символьное устройство

Навигация по файловой системе является одним из самых важных навыков при работе с операционной системой Linux. Основными командами, используемыми при навигации по файловой системе, являются:

Таблица 4.2. Основные утилиты.

Команда	**Описание**
pwd	показывает полное имя каталога
cd	изменяет текущий каталог на указанный
pushd, popd	изменяет каталог на указанный
cp	используется для копирования файлов
touch	позволяет создавать файлы
rm	используется для удаления файлов
mkdir	позволяет создать каталог
rmdir	позволяет удалить каталог

Файловая система Linux, как и любой другой unix-подобной операционной системы, имеет строгую структуру каталогов. Каждый дистрибутив Linux может несколько изменять структуру в зависимости от предпочтений разработчиков. Мы рассмотрим те каталоги, которые используются в каждом дистрибутиве:

Таблица 4.3 Основные каталоги.

Каталог	**Описание**
/bin	в этом каталоге находятся основные исполняемые файлы, необходимые для функционирования системы

/boot	содержит ядро операционной системы и карты загрузки, а также конфигурационные файлы загрузчиков
/dev	содержит файлы, которые являются интерфейсом с периферийными устройствами
/etc	содержит основные файлы настроек приложений Linux
/home	содержит домашние папки пользователей
/lib	содержит основные библиотеки, необходимые для нормальной работы системы
/lost+found	информация, восстановленная при проверке файловой системы на наличие ошибок
/media	точки монтирования отключаемых устройств (usb-диски, CD, floppy)
/mnt	точки монтирования ISO-образов, сетевых файловых систем, других постоянных файловых систем
/opt	альтернатива usr, для коммерческого ПО или ПО, не входящего в основной дистрибутив
/proc	содержит общую информацию о системе и подробную о процессах.
/root	домашний каталог пользователя root
/sbin	утилиты суперпользователя

/srv	файлы, выкладываемые для доступа всевозможных внешних служб
/sys	внутри этого каталога также находится виртуальная файловая система, только она содержит подробную информацию о процессах
/tmp	в этом каталоге находятся временные файлы, используемые запущенными в данный момент процессами
/usr	программы, библиотеки и другие данные пользовательских приложений
/var/log	содержит файлы журналов

4.1.3. Учетные записи ОС Linux

Linux, как и любая unix-подобная система, является не только многозадачной, но и многопользовательской, т.е. эта операционная система позволяет одновременно нескольким пользователям работать с ней. Но система должна как-то узнавать, какой или какие из пользователей работают в данный момент. Именно для этих целей в Linux существует два понятия – учетные записи и аутентификация, которые являются частями одного механизма.

Учетная запись пользователя – это необходимая для системы информация о пользователе, хранящаяся в специальных файлах. Информация используется Linux для аутентификации пользователя и назначения ему прав доступа.

Аутентификация – системная процедура, позволяющая Linux определить, какой именно пользователь осуществляет вход.

Вся информация о пользователе обычно хранится в файлах /etc/passwd и /etc/group.

/etc/passwd – этот файл содержит информацию о пользователях. Запись для каждого пользователя занимает одну строку:

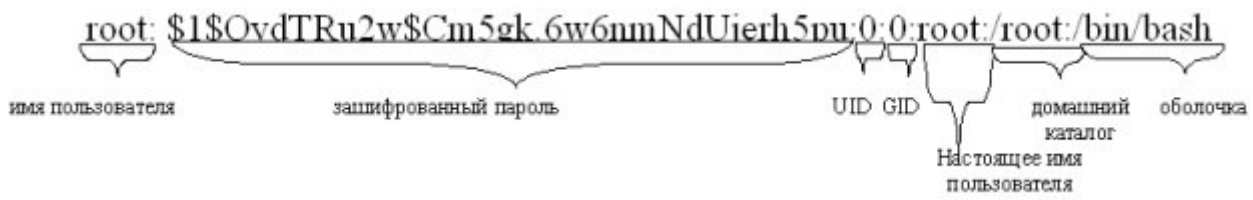

Рисунок 4.1. Информация о пользователе.

Имя пользователя – имя, используемое пользователем на все приглашения типа login при аутентификации в системе.

Зашифрованный пароль – обычно хешированный по необратимому алгоритму MD5 пароль пользователя или символ '!', в случаях, когда интерактивный вход пользователя в систему запрещен.

UID – числовой идентификатор пользователя. Система использует его для распределения прав файлам и процессам.

GID – числовой идентификатор группы. Имена групп расположены в файле /etc/group. Система использует его для распределения прав файлам и процессам.

Настоящее имя пользователя – используется в административных целях, а также командами типа finger (получение информации о пользователе через сеть).

Домашний каталог – полный путь к домашнему каталогу пользователя.

Оболочка – командная оболочка, которую использует пользователь при сеансе. Для нормальной работы она должна быть указана в файле регистрации оболочек /etc/shells.

/etc/group – этот файл содержит информацию о группах, к которым принадлежат пользователи:

Рисунок 4.2. Информация о группах.

Имя группы – имя, применяемое для удобства использования таких программ, как newgrp.

Шифрованный пароль – используется при смене группы командой newgrp. Пароль для групп может отсутствовать.

GID – числовой идентификатор группы. Система использует его для распределения прав файлам и процессам.

Пользователи, включенные в несколько групп – В этом поле через запятую отображаются те пользователи, у которых по умолчанию (в файле /etc/passwd) назначена другая группа.

На сегодняшний день хранение паролей в файлах passwd и group считается ненадежным. В новых версиях Linux применяются так называемые теневые файлы паролей – shadow и gshadow. Права на них назначены таким образом, что даже чтение этих файлов без прав суперпользователя невозможно. Нужно учесть, что нормальное функционирование системы при использовании теневых файлов подразумевает одновременно и наличие файлов passwd и group. При использовании теневых паролей в /etc/passwd и /etc/group вместо самого пароля устанавливается символ 'x', что и является указанием на хранение пароля в /etc/shadow или /etc/gshadow.

Файл shadow хранит защищенную информацию о пользователях, а также обеспечивает механизмы устаревания паролей и учетных записей. Структура файла shadow, где:

а - имя пользователя;

б - шифрованный пароль – применяются алгоритмы хеширования, как правило MD5 или символ '!', в случаях, когда интерактивный вход пользователя в систему запрещен;

в - число дней с последнего изменения пароля, начиная с 1 января 1970 года;

г - число дней, перед тем как пароль может быть изменен;

д - число дней, после которых пароль должен быть изменен;

е - число дней, за сколько пользователя начнут предупреждать, что пароль устаревает;

ж - число дней, после устаревания пароля для блокировки учетной записи;

з -дней, отсчитывая с 1 января 1970 года, когда учетная запись будет заблокирована;

и - зарезервированное поле.

Рисунок 4.3. Структура файла shadow.

Файл gshadow так же накладывает дополнительную функциональность, вкупе с защищенным хранением паролей групп.

Рисунок 4.4. Структура файла gshadow.

Имя группы – имя, используемое для удобства использования таких программ, как newgrp.

Шифрованный пароль – используется при смене группы командой newgrp . Пароль для групп может отсутствовать.

Администратор группы – пользователь, имеющий право изменять пароль с помощью gpasswd.

Список пользователей – В этом поле через запятую отображаются те пользователи, у которых по умолчанию (в файле /etc/passwd) назначена другая группа.

В Linux, кроме обычных пользователей, существует один (и только один) пользователь с неограниченными правами. Идентификаторы UID и GID такого пользователя всегда 0. Его имя, как правило, root , однако оно может быть легко изменено (или создано несколько символьных имен с одинаковым GID и UID), так как значение для применения неограниченных прав доступа имеет только GID 0 . Для пользователя root права доступа к файлам и процессам не проверяются системой. При работе с использованием учетной записи root необходимо быть предельно осторожным, т.к. всегда существует возможность уничтожить систему.

В Linux используется развитая система распределения прав пользователям. Но для точного опознания пользователя одного имени недостаточно с точки зрения безопасности. Именно поэтому используется и пароль – произвольный набор символов произвольной длины, обычно ограниченной лишь используемыми методами шифрования.

Сегодня в большинстве версий Linux пароли шифруются по алгоритмам 3DES и MD5 (устарело, теперь SHA512). Когда алгоритм 3DES является обратимым, то есть такой пароль можно расшифровать, MD5 – это необратимое преобразование. Пароли, зашифрованные по алгоритму 3DES не применяются при использовании теневых файлов для хранения паролей.

При аутентификации, пароль, введенный пользователем, шифруется тем же методом, что и исходный, а потом сравниваются уже зашифрованные копии. Если они одинаковые, то аутентификация считается успешной.

Учитывая ежедневно увеличивающиеся требования к безопасности, в Linux есть возможность использовать скрытые пароли. Файлы /etc/passwd и /etc/group доступны для чтения всем пользователям, что является довольно большой брешью в безопасности системы. Именно поэтому в современных версиях Linux предпочтительнее использовать скрытые пароли. Такие пароли располагаются в файлах /etc/shadow и /etc/gshadow, для паролей пользователей и групп соответственно.

Команда login запускает сеанс интерактивной работы в системе. Она проверяет правильность ввода имени и пароля пользователя, меняет каталог на домашний, выстраивает окружение и запускает командный интерпретатор. Команду login как правило не запускают из командной строки — это обычно за пользователя делают менеджеры консоли — например getty или mgetty.

Команда su (switch user) позволяет сменить идентификатор пользователя уже в процессе сеанса. Синтаксис ее прост: su username, где username – имя пользователя, которое будет использоваться. После этого программа запросит пароль. При правильно введенном пароле, su запустит новый командный интерпретатор с правами пользователя, указанного su и присвоит сеансу его идентификаторы. Если имя пользователя опущено, то команда su использует имя root.

```
[student@ns student]$ su root
Password:
[root@ns student]#_
```

Листинг 4.2.

При использовании команды su пользователем root она, как правило, не запрашивает пароль.

Команда newgrp аналогична по своим возможностям su с той разницей, что происходит смена группы. Пользователь должен быть включен в группу, которая указывается в командной строке newgrp . При использовании команды newgrp пользователем root она никогда не запрашивает пароль. Синтаксис команды аналогичен синтаксису команды su: newgrp groupname, где groupname – имя группы, на которую пользователь меняет текущую.

Команда passwd является инструментом для смены пароля в Linux. Для смены своего пароля достаточно набрать в командной строке passwd:

```
[student@ns student]$ passwd
Changing password for student
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully
[student@ns student]$_
```

Листинг 4.3.

Для смены пароля группы и управления группой используется команда gpasswd Для смены пароля достаточно набрать в командной строке gpasswd GROUPNAME. Сменить пароль вам удастся только если Вы являетесь администратором группы. Если пароль не пустой, то для членов группы вызов newgrp пароля не требует, а не члены группы должны ввести пароль. Администратор группы может добавлять и удалять пользователей с помощью параметров -a и -d соответственно. Администраторы могут использовать параметр -r для удаления пароля группы. Если пароль не задан, то только члены группы с помощью команды newgrp могут войти в группу. Указав параметр -R можно запретить доступ в группу по паролю с помощью команды newgrp (однако на членов группы это не распространяется). Системный администратор (root) может использовать параметр -A, чтобы назначить группе администратора.

Команда chage управляет информацией об устаревании пароля и учетной записи. Обычный пользователь (не root) может использовать команду только для просмотра своих параметров устаревания пароля:

```
gserg@ADM:/$ chage -l gserg
Last password change                                    : Май 03,
2007
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

Листинг 4.4.

Суперпользователь же может использовать также иные параметры, такие как:

-d дата (в формате системной даты, например ДД.ММ.ГГГГ) – устанавливает дату последней смены пароля пользователем.

-E дата – установить дату устаревания пароля учетной записи пользователя

-I N – установить количество дней неактивности N с момента устаревания пароля перед тем, как учетная запись будет заблокирована

-m N – задает минимальное количество дней (N) между сменами пароля

-M N – задает максимальное количество дней (N) между сменами пароля

-W N – задает количество дней, за которые будет выдаваться предупреждение об устаревании пароля.

4.1.4. Права доступа

Для каждого объекта в файловой системе Linux существует набор прав доступа, определяющий взаимодействие пользователя с этим объектом. Такими объектами могут быть файлы, каталоги, а также специальные файлы (например, устройства) — то есть, по сути, любой объект файловой системы. Так как у каждого объекта в Linux имеется владелец, то права доступа применяются относительно владельца файла. Они состоят из набора 3 групп по три атрибута:

– чтение (r), запись (w), выполнение (x) для владельца;

– чтение, запись, выполнение для группы владельца;

– чтение, запись, выполнение для всех остальных.

Такие права можно представить краткой записью:

rwxrwxrwx – разрешено чтение, запись и выполнение для всех

rwxr-xr-x – запись разрешена только для владельца файла, а чтение и выполнение для всех.

rw-rw-r-- – запись разрешена для владельца файла и группы владельца файла, а чтение – для всех.

Такое распределение прав позволяет гибко управлять ресурсами, доступными пользователям.

Права доступа распространяются и на каталоги. Они означают:

r – если установлено право на чтение из каталога, то можно увидеть его содержимое командой ls.

w – если установлено право записи в каталог, то пользователь может создавать и удалять файлы из текущего каталога. Причем удалить файл из каталога пользователь может даже если у него нет прав на запись в файл. Есть возможность исправить эту ситуацию. Об этом я скажу позже.

x – если установлено право исполнения на каталог, то пользователь имеет право перейти в такой каталог командами наподобие cd.

Таким образом появляется возможность создания так называемых "скрытых" каталогов, когда невозможно получить список файлов, но пользователь, точно знающий имя файла, может скопировать его из "скрытого" каталога.

Для распределения прав доступа в Linux существует множество команд. Основные из них – это chmod, chown и chgrp.

Команда chmod (Change MODe – сменить режим) – изменяет права доступа к файлу. Для использования этой команды также необходимо иметь права владельца файла или права root . Синтаксис команды таков:

chmod mode filename, где

filename – имя файла, у которого изменяются права доступа;

mode – права доступа, устанавливаемые на файл. Права доступа можно записать в 2 вариантах – символьном и абсолютном.

В символьном виде использование команды chmod будет выглядеть следующим образом:

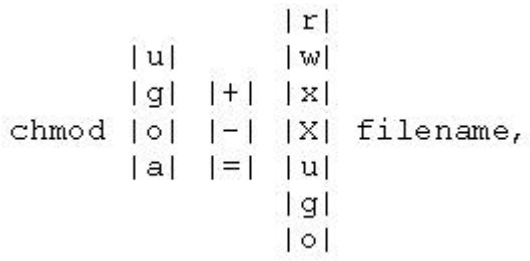

Рисунок 4.5. Символьное представление команды chmod.

где:

u, g, o, a – установка прав для пользователя, группы, остальных пользователей, всех групп прав доступа соответственно.

+, -, = – добавить, удалить, установить разрешение соответственно.

r, w, x, X, u, g, o – право чтения, записи, выполнения, выполнения если есть такое право еще у какой-либо из групп доступа, такие же как у владельца, такие же как у группы, такие же как у остальных пользователей.

filename - Имя файла, у которого изменяются права.

Просмотр разрешений, установленных на файл, осуществляется командой ls с ключом -l:

```
[student@ns student]$ ls -l lesson5.txt
-rw-------  1  student student  39 Nov 19 15:17  lesson5.txt
[student@ns student]$ chmod g+rw lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw----  1  student student  39 Nov 19 15:18  lesson5.txt
[student@ns student]$ chmod o=u lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw-rw-  1  student student  39 Nov 19 15:18  lesson5.txt
[student@ns student]$ chmod o-w lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rw-rw-r--  1  student student  39 Nov 19 15:19  lesson5.txt
[student@ns student]$
```

Листинг 4.5.

Для использования абсолютного режима необходимо представить права доступа к файлу в виде 3-х двоичных групп. Так например:

rwx r-x r-- будет выглядеть как: 111 101 100

Теперь каждую двоичную группу перевести в 8-ричное число: 111 – 7, 101 – 5, 100 – 4.

Чтобы задать файлу такие права необходимо выполнить команду:

```
[student@ns student]$ ls -l lesson5.txt
-rw-rw-r--  1  student student  39 Nov 19 15:19  lesson5.txt
[student@ns student]$chmod 754 lesson5.txt
[student@ns student]$ ls -l lesson5.txt
-rwxr-xr--  1  student student  39 Nov 19 15:19  lesson5.txt
[student@ns student]$
```

Листинг 4.6.

Задание для обучаемых: попробовать изменить права файлу lesson5.txt и задать следующие: r w x r- - r - - (744), r - - - w- - - x(421), - - x - w - r - -(124).

Также предложить им проделать то же самое в символьном виде.

Команда chown (CHange OWNer – сменить владельца) – позволяет сменить владельца файла. Для использования этой команды необходимо либо иметь права владельца текущего файла или права root. Синтаксис команды прост:

chown username:groupname filename , где

username – имя пользователя – нового владельца файла;

groupname – имя группы – нового владельца файла;

filename – имя файла, у которого сменяется владелец.

Имя группы в синтаксисе команды можно не указывать, тогда будет изменен только владелец файла.

Команда chgrp используется для изменения владельца-группы файла. Синтаксис ее таков:

chgrp groupname filename,

где:

groupname – имя группы, которой будет принадлежать файл

filename – имя изменяемого файла

Имейте в виду, что использовать команды chown и chmod может только пользователь-владелец файла и root , а команду chgrp - пользователь-владелец файла, группа-владелец файла и root.

Существуют еще несколько особых прав, которые могут устанавливаться на файлы и каталоги. О некоторых из них мы поговорим при изучении темы "процессы". Но один рассмотрим сейчас. Это так называемый sticky bit (бит прикрепления).

В первых версиях Юникс этот бит использовался для того, чтобы заставить систему при работе программы оставлять образ ее кода в памяти. Тогда при следующем обращении к программе на ее запуск тратилось намного меньше времени так как чтение кода с устройства более не требовалось. Для файлов и сегодня в Linux осталось прежнее значение этого бита. А вот для каталогов этот атрибут приобрел новое значение. Если sticky bit установлен на каталог, то удалить файлы из такого каталога может только пользователь-владелец файла, и то только если у него есть право на запись в файл. Группа-владелец и остальные пользователи даже при наличии прав на запись в файл не смогут удалить его при установленном на каталог sticky bit.

Бит прикрепления устанавливается командой chmod в символьном виде:

chmod +t filename

4.2. Концепции безопасности Unix

В операционной системе UNIX используется достаточно постая модель доступа, основанная на субъект-субъектной модели. В современных версиях UNIX помимо общей схемы можно использовать списки доступа. При этом реализуется статическая авторизация множественного доступа к объекту.

Пользователи и группы.

В UNIX роль номинального субъекта безопасности играет пользователь. Каждому пользователю выдается (обычно – одно) входное имя (login). Каждому входному имени соответствует единственное число, идентификатор

пользователя (User IDentifier, UID). Это число и есть ярлык субъекта, которым система пользуется для определения прав доступа.

Каждый пользователь входит в одну или более групп. Группа – это образование, которое имеет собственный идентификатор группы (Group IDentifier, GID), объединяет нескольких пользователей системы, а стало быть, соответствует понятию множественный субъект. Значит, GID – это ярлык множественного субъекта, каковых у действительного субъекта может быть более одного. Таким образом, одному UID соответствует список GID.

Роль действительного (работающего с объектами) субъекта играет процесс. Каждый процесс снабжен единственным UID: это идентификатор запустившего процесс пользователя. Любой процесс, порожденный некоторым процессом, наследует его UID. Таким образом, все процессы, запускаемые по желанию пользователя, будут иметь его идентификатор. UID учитываются, например, когда один процесс посылает другому сигнал. В общем случае разрешается посылать сигналы «своим» процессам (тем, что имеют такой же UID).

Права доступа.

Роль объекта в UNIX играют многие реальные объекты, в частности представленные в файловой системе: файлы, каталоги, устройства, каналы и т. п.. Каждый файл снабжён UID – идентификатором пользователя-владельца. Вдобавок у файла есть единственный GID, определяющий группу, которой он принадлежит.

На уровне файловой системы в UNIX определяется три вида доступа: чтение (read, r), запись (write, w) и использование (execution, x). Право на чтение из файла дает доступ к содержащейся в нем информации, а право записи – возможность ее изменять. При каждом файле имеется список того, что с ним может делать владелец (если совпадает UID процесса и файла), член группы владельцев (если совпадает GID) и кто угодно (если ничего не совпадает) (см. Рисунок 4.6. «Базовые права доступа в UNIX»). Такой список для каждого объекта системы занимает всего несколько байт.

Рисунок 4.6. Базовые права доступа в UNIX.

Флаг использования трактуется по-разному в зависимости от типа файла. В случает простого файла он задаёт возможность исполнения файла, т.е. запуска программы, содержащейся в этом файле. Для директории – это возможность доступа к файлам в этой директории (точнее говоря, к атрибутам этих файлов – имени, правам доступа и т.п..).

Рассмотрим последовательность проверки прав на примере (см. Рисунок 4.7, «Последовательнось проверки прав доступа в UNIX»). Пусть файл имеет следующие атрибуты:

file.txt alice:users rw- r-- ---

Т.е. файл принадлежит пользователю «alice», группе «users» и имеет права на чтение и запись для владельца и только чтение для группы.

Пусть файл пытается прочитать пользователь «bob». Он не является владельцем, однако он является членом группы «users». Значит, он имеет права на чтение этого файла.

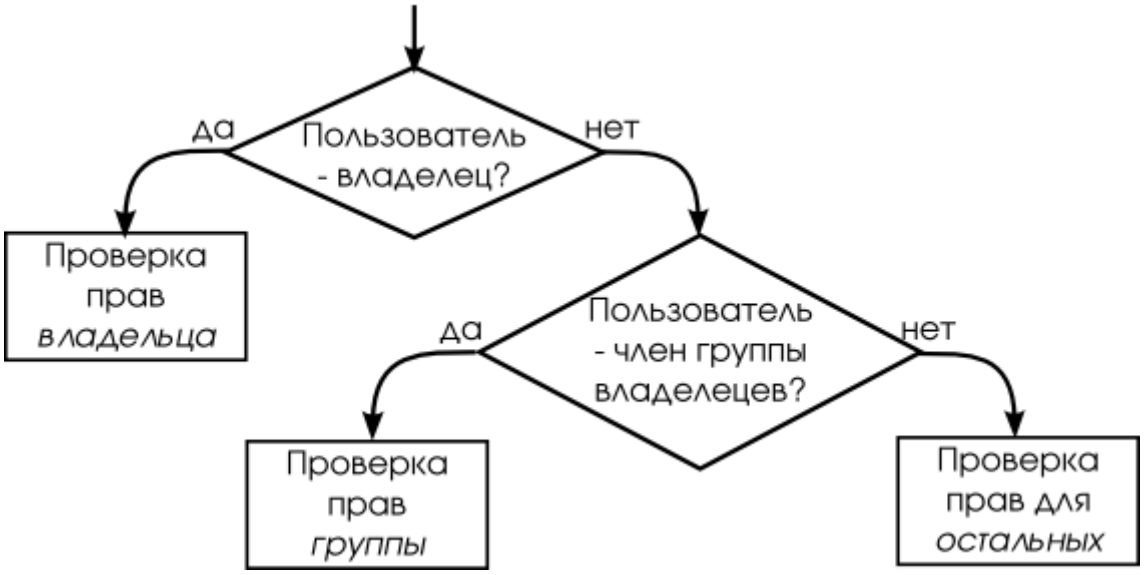

Рисунок 4.7. Последовательность проверки прав доступа в UNIX.

Разделяемые каталоги

Права записи в директорию трактуются как возможность создания и удаления файлов, а также изменение атрибутов файлов (например, переименование). При этом субъекту не обязательно иметь права на запись в эти удаляемые файлы.

Таким образом, из своего каталога пользователь может удалить любой файл. А если запись в каталог разрешена всем, то любой пользователь сможет удалить в нём любой файл. Для избегания этой проблемы был добавлен ещё один бит в права доступа каталога: *бит навязчивости (sticky, t-бит)*. При его установке пользователь, имеющий доступ на запись в этот каталог, может изменять только *собственные* файлы.

Подмена идентификатора субъекта.

В UNIX существует механизм, позволяющий пользователям запускать процессы от имени других пользователей. Это может быть полезным, если одному пользователю необходимо на время предоставлять права другого (например, суперпользователя).

Для разрешения подмены идентификатора пользователя применяется бит подмены идентификатора пользователя (set user id, suid-бит, s). Этот бит

применяется совместно с битом исполнения (x) для обычных файлов. При установке этого бита на исполняемый файл процесс запускается от имени владельца, а не от имени запускающего пользователя (см. Рисунок 4.8, «Подмена идентификатора пользователя»).

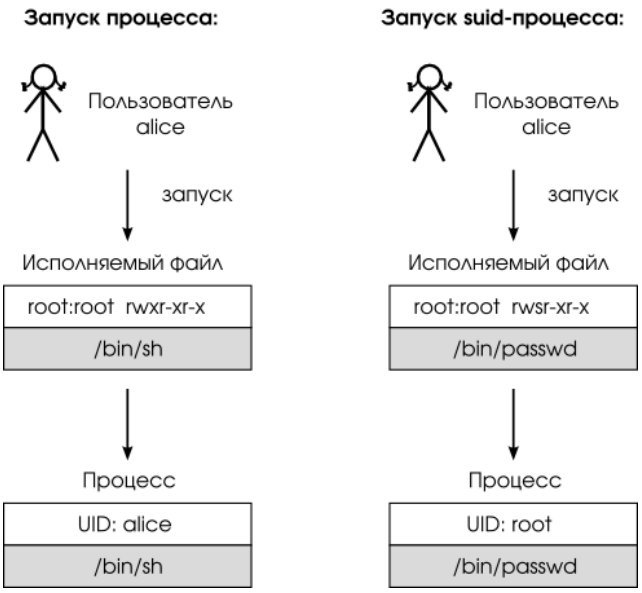

Рисунок 4.8. Подмена идентификатора пользователя.

Подмена идентификатора пользователя является потенциальной угрозой безопасности системы и должна использоваться осторожно.

Недостатки базовой модели доступа и её расширения.

Ограниченность системы прав UNIX приводит к тому, что, к примеру, невозможно создать такое положение вещей, когда одна группа пользователей могла бы только читать из файла, другая – только запускать его, а всем остальным файл вообще не был бы доступен. Другое дело, что такое положение вещей встречается нечасто.

Со временем в различных версиях UNIX стали появляться расширения прав доступа, позволяющие устанавливать права на отдельные объекты системы. Поначалу это были так называемые флаги – дополнительные атрибуты файла, не позволяющие, например, переименовывать его или удалять из него информацию при записи (можно только дописывать). Флаги не устраняют главного

недостатка, зато их легко организовать без изменения файловой системы: каждый флаг занимает ровно один бит.

Многие современные файловые системы UNIX поддерживают также списки доступа (ACL), с помощью которых можно для каждого объекта задавать права всех субъектов на доступ к нему.

На практике флаги или управление доступом использовать приходится нечасто. В большинстве случаев такая необходимость возникает в виде исключения – например, для временного уменьшения прав или для временного предоставления доступа (легко сделать с помощью ACL), а также при работе с очень важными файлами.

Суперпользователь.

Пользователь root (он же суперпользователь) имеет нулевые UID и GID и играет роль доверенного субъекта UNIX. Это значит, что он не подчиняется законам, которые управляют правами доступа, и может по своему усмотрению эти права изменять. Большинство настроек системы доступны для записи только суперпользователю.

В UNIX существует уровень доступа ядра и уровень доступа системы. Суперпользователь работает на уровне доступа ядра, так что является по сути продолжением самой системы.

Многие команды должны исполняться только от имени суперпользователя, так как в них производится взаимодействие с частями ядра, отвечающими за взаимодействие с аппаратурой, права доступа и т.п.. Если же такие команды разрешается запускать простым пользователям, применяется рассмотренный выше механизм подмены идентификатора пользователя.

Системное администрирование в UNIX производится от имени пользователя *root*. При работе от этого имени следует быть очень осторожным: выполнение неверной команды может привести к краху системы и уничтожению информации. Поэтому даже администраторы никогда не работают в сеансе суперпользователя всё время, а переходят в режим суперпользователя только тогда, когда это действительно необходимо (например, с помощью команды *su*).

Аутентификация пользователей.

В UNIX сеанс работы пользователя начинается с его аутентификации и заканчивается его выходом из системы. При входе в систему выполняется следующая последовательность действий (см. Рисунок 4.9, «Процесс входа в систему»):

– процесс getty ожидает реакции пользователя на одной из терминальных линий, в случае активности пользователя выводит приглашение;

– после ввода имени пользователя запускается программа login, которая проверяет подлинность пользователя. Стандартным механизмом является проверка пароля, заданного для данного пользователя;

– убедившись, что пароль введён правильно, login запускает командный интерпретатор с установленными UID и GID данного пользователя. Таким образом, права доступа любой программы (действительного субъекта), запущенной пользователем в этом сеансе работы, будут определяться правами номинального субъекта UID+GID.

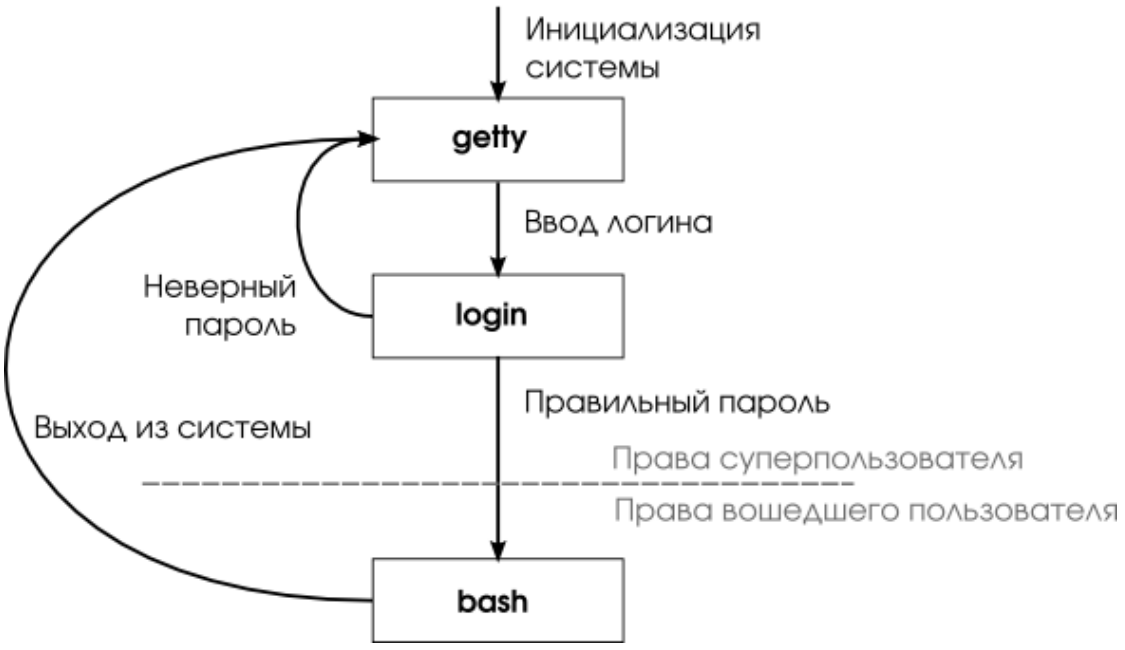

Рисунок 4.9. Процесс входа в систему.

При работе по сети роль getty исполняет сетевой демон, например ssh.

В некоторых современных UNIX-системах существуют расширения систем авторизации и аутентификации. Например, в Linux-системах этот механизм называется подключаемые модули аутентификации (Pluggable Authentication Modules, PAM). Эти средства выходят за рамки данных лекций.

4.2.1. Настройка системы безопасности

База данных пользователей системы.

Все данные о пользователях UNIX хранит в файле /etc/passwd в текстовом виде. Каждому пользователю соответствует одна строка, поля которой разделяются двоеточиями:

входное имя:x:UID:GID:полное имя:домашний каталог:стартовый shell

Каждый пользователь явно связан с одной из групп – это основная группа пользователя. Это сделано для того, чтобы каждый пользователь состоял хотя бы в одной группе. Все новые файлы, создаваемые этим пользователем, в качестве группы владельцев будут иметь его основную группу.

Из примера видно, что некоторые пользователи имеют «неправильные» командные оболочки, работа в которых невозможна. Это сделано специально для того, чтобы исключить возможность входа таких пользователей в систему.

Пароли на вход в систему пользователей в UNIX не хранятся в открытом виде, хранятся только их хэши (набор байт, получаемый из пароля с помощью односторонней функции). Даже если злоумышленник получит значение этого хэша, ему придется подбирать пароль, применяя данную одностороннюю функцию к различным словам и сравнивая со значением хэша. Часто хэши хранятся в спациальном файле (например, /etc/shadow), доступ к которому разрешен только системе, так что перебор вообще невозможен.

Аналогичным образом информация о группах хранится в файле /etc/group. Каждой строке файла соответствует информация о группе: её имя, числовой идентификатор и список пользователей, входящих в эту группу.

Изменение информации о пользователях, также как и добавление новых пользователей, может производиться простым редактированием этого файла, однако более корректным способом является использование специальных утилит, которые рассматриваются далее.

Изменение базы данных пользователей.

Для добавления и удаления пользователей и групп существует набор команд: useradd, userdel, groupadd, groupdel. Эти команды доступны только суперпользователю и имеют единственный обязательный параметр: имя пользователя или группы.

С помощью команд usermod и groupmod можно изменять информацию в базах данных пользователей и групп. Эти команды также может выполнять только администратор системы.

Команда passwd позволяет простым пользователям изменять свой системный пароль, а суперпользователю – изменять пароль любого из пользователей системы.

Изменение прав доступа.

Для изменения владельца файла или группы владельцев используются команды chown и chgrp. Из соображений безопасности спользовать эти команды может только суперпользователь.

Пользователь может изменять права доступа к своим файлам с помощью команды chmod.

Ограничения сеанса пользователя.

В UNIX существует ряд динамических ограничений, накладываемых на процесс аутентификации пользователя и запущенные им программы. Ограничения можно разделить на следующие группы:

ограничения входа в систему

Вход пользователя в систему может быть ограничен видом терминала, удалённым адресом (в случае сетевого входа в систему), временем работы. Для задания этих ограничений в некоторых UNIX-системах используется файл /etc/login.access.

ограничения запускаемых процессов

Процессы пользователей могут быть ограничены по одному из следующих параметром: объём используемой памяти, число одновременно открытых файлов, число запускаемых процессов и т.п. Для задания этих ограничений в некоторых UNIX-системах используется файл /etc/limits.

ограничения использования диска

Дисковые квоты позволяют ограничить объём используемого пространства жёсткого диска для каждого из пользователей системы. Для настройки данного ограничения необходима утилита quote, а также поддержка в выбранной файловой системе.

Ограничения действуют на протяжении всего сеанса работы пользователя.

4.2.2. Основы информационной безопасности UNIX

Информационная безопасность – меры по защите информации от неавторизованного доступа, разрушения, модификации, раскрытия и задержек в доступе. Основой информационной безопасности любой организации является политика безопасности.

Политика безопасности

Политика безопасности – это набор законов, правил и норм поведения, определяющих, как организация обрабатывает, защищает и распространяет информацию. Это активный компонент защиты, который включает в себя анализ возможных угроз и выбор мер противодействия.

Важным элементом политики безопасности является управление доступом: ограничение или исключение несанкционированного доступа к информации и программным средствам. При этом используются два основных понятия: объект и субъект системы. Объектом системы мы будем называть любой её идентифицируемый ресурс (например, файл или устройство). Доступом к объекту системы – некоторую заданную в ней операцию над этим объектом (скажем, чтение или запись). Действительным субъектом системы

назовем любую сущность, способную выполнять действия над объектами (имеющую к ним доступ). Действительному субъекту системы соответствует некоторая абстракция, на основании которой принимается решение о предоставлении доступа к объекту или об отказе в доступе. Такая абстракция называется номинальным субъектом. Например, курсант КрУ МВД России– действительный субъект, его пропуск в КрУ МВД России – номинальный. Другим примером может служить злоумышленник, прокравшийся в секретную лабораторию с украденной картой доступа – он является действительным субъектом, а карта – номинальным (см. Рисунок 2.5, «Объект и субъект безопасности»).

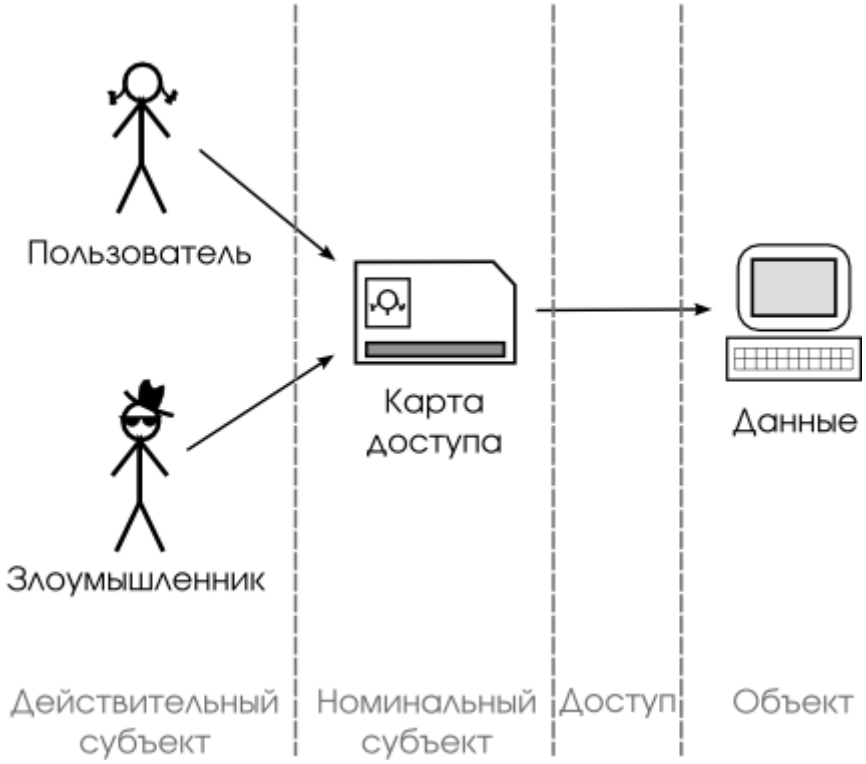

Рисунок 4.10. Объект и субъект безопасности.

Политика безопасности должна быть *полной, непротиворечивой* и рассматривать все возможности доступа субъектов системы к её объектам. Только соблюдение всех трех принципов гарантирует, что нарушить установленные правила (например, получить несанкционированный доступ к объекту) системными средствами невозможно. Если же предполагаемый

злоумышленник воспользовался каким-нибудь внесистемным средством и смог получить статус номинального субъекта, к которому он не имеет отношения (например, подглядел чужой пароль и работает под чужим именем), никаких гарантий быть не может.

Полнота политики безопасности означает, что в ней должны быть отражены все существующие ограничения доступа. Непротиворечивость заключается в том, что решение об отказе или предоставлении доступа конкретного субъекта к конкретному объекту не должно зависеть от того, какими путями система к нему приходит. Третье требование, называемое также отсутствием недокументированных возможностей, должно гарантировать нам, что доступ не может быть осуществлен иначе как описанным в политике безопасности способом.

Политика безопасности включает в себя технические, организационные и правовые аспекты, в рамках этих лекций рассматривается только технический аспект.

Управление доступом

Существует несколько схем управления доступом, называемых моделями доступа. Рассмотрим самые известные из них:

Мандатная модель доступа

Объектам и субъектам системы ставится в соответствие метка безопасности или мандат (например, гриф секретности). При этом метка безопасности субъекта описывает его благонадёжность, а метка безопасность объекта – степень закрытости информации. Доступ к объекту разрешён только субъектам с соответствующей или более сильной меткой.

Недостатком такой схемы можно считать слишком грубое деление прав, необходимость введения категорий доступа и т.п. Также для данной модели доступа очень важно разработать механизм понижения секретности теряющих важность документов.

Списки доступа (Accecc Control Lists, ACL)

Все субъекты и объекты системы объединяются в таблицу, в строках которой находятся субъекты (активные сущности), а в столбцах – объекты (пассивные сущности), элементы же такой таблицы содержат перечисление прав, которыми субъект обладает в отношении данного объекта. Такая схема называется субъект-объектная модель.

Недостатками можно считать огромный размер таблицы и сложность администрирования в случае большого числа объектов и субъектов в системе.

Произвольное управление доступом

Каждому объекту сопоставляется один субъект – владелец объекта. Владелец может по своему усмотрению давать другим субъектам или отнимать у них права на доступ к объекту. Если объект имеет несколько хозяев, они могут быть объединены общим субъектом – группой. Такая схема позволяет значительно сократить размер таблицы прав субъектов по отношению к объектам. Эта схема также называется субъект-субъектная модель.

Недостатком этой схемы является значительное облегчение управления доступом, что не позволяет простроить сложные отношения между субъектами и объектами.

Аутентификация и авторизация

Авторизация – это процесс определения того, имеет или не имеет некоторый субъект доступ к некоторому объекту. Авторизация может быть:

статической

вопрос о доступе к объекту решается один раз, когда права задаются или изменяются, при этом пользователю ставится в соответствие некоторый номинальный субъект системы;

динамической

принятие решения о доступе производится при каждом обращении к объекту, часто это носит характер ограничения возможностей пользователя по объёму памяти и дискового пространства, времени работы и т.п.

Процессу *авторизации* всегда должен предшествовать процесс аутентификации. Аутентификация – это механизм сопоставления работающего пользователя системы некоторому номинальному субъекту. Как правило, при этом пользователю необходимо ввести пароль или предоставить секретный ключ.

4.3. Разграничение прав доступа пользователей

4.3.1. Права администратора

Ограничение прав пользователей в операционной системе – тема сложная, но необходимая. В зависимости от семейства операционной системы существуют разные правила и стандарты. Рассмотрим подробнее, как это происходит в ОС Ubuntu.

Чтобы запускать привилегированные команды, пользователь должен обладать правами, как минимум, администратора системы. По умолчанию операционная система отключает повышенный уровень прав у любого пользователя. Чтобы его повысить, необходимо использовать следующей командой.

usermod -a -G sudo username1

Теперь пользователь с именем username1 добавлен в группу sudo и является администратором для операционной системы. Ему доступны настройки ОС, а также доступ к каталогу /dev с вложениями. Большинство привилегий администраторов схоже с возможностями суперпользователя, но они неполные.

В корпоративных информационных системах большинство ОС являются многопользовательскими. Соответственно, необходимо для каждого владельца разграничить права доступа. Для этого используется внутренняя команда chmod, например:

sudo chmod o-x $(which ls)>

Вышеуказанная строка означает, что только root имеет право запускать команду ls. Всем остальным в доступе будет отказано.

Разберем другую ситуацию. Есть пользователь с именем username1. Ему необходимо ограничить доступ к команде ls. Для этого создаем группу пользователей usergroup1, в которую перенесём всех кроме username1.

sudo groupadd usergroup1
sudo useradd -G usergroup1 <username2, username3>

Вторая строка добавляет в группу usergroup1 пользователей username2, username3 и т. д. Ограничим права на запуск команды ls. Её смогут активировать только участники usergroup1.

sudo chown :group2 $(which ls)
sudo chmod 754 $(which ls)

Теперь неучастник usergroup1 не сможет активировать ls.

4.3.2. Файл /etc/sudoers

Sudoers содержит информацию о пользователях, которые могут использовать утилиту sudo. Чтобы открыть файл, используем специальную утилиту visudo.

visudo /etc/sudoers

Внутри содержится информация представленная на рисунке 4.11.

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

Рисунок 4.11. Содержимое sudoers.

Важной информацией содержимого suoders является строка %sudo ALL = (ALL:ALL) ALL. %sudo означает, что к группе sudo применяется следующее правило. Если устанавливаем правила для конкретного пользователя, то % не нужен. Первая переменная ALL расшифровывает, как применить правило ко всем IP-адресам. Второй и третий ALL – указанный пользователь или группа имеют право исполнять команды в сессии любого пользователя или группы. Четвертая переменная означает, что данный шаблон применяется ко всем командам. Например, необходимо установить права на запуск утилиты apt get для группы admin.

%admin ALL=(ALL)NOPASSWD:/usr/bin/apt-get

4.3.3. Alias (псевдонимы)

Для удобства разграничения прав доступа используются алиасы. Они объединяют один или несколько значений в один параметр. Например, присвоим IP-адресу облачного хранилища более удобное имя.

Host_Alias CLOUD = 105.17.125.37

CLOUD – псевдоним, который указывается в параметрах вместо IP-адреса.

При необходимости алиасы используются для объединения пользователей в одну группу.

User_Alias Name = user1,user2,...

Name – псевдоним, а user1, user2 – имена пользователей. Также утилита Alias доступна и для команд, т. е. объединяем список инструкций в единую группу.

Cmnd_Alias Name = cmd1,cmd2

name – произвольное наименование для перечня команд;

cmd1, cmd2 – перечисление команд через запятую.

Пример объединения обновления пакетов в алиас: *cmnd_Alias APT = /usr/bin/apt-get update,/usr/bin/apt-get upgrade.*

4.3.4. Права пользователей в Linux. Команды sudo и su

Linux — сложная система, поддерживающая многопользовательский режим. Это утверждение означает, что в режиме реального времени одновременно параллельно с системой может работать множество человек, и каждый из них способен запустить несколько приложений. В Linux настроить права пользователя совсем несложно.

Root обладает в указанной системе максимальными полномочиями, и она полностью подвластна ему, — любая его команда будет выполнена сразу. Поэтому работать таким доступом следует с осторожностью.

Рекомендуется работать под стандартным пользователем, а максимальные привилегии получать только тогда, когда это действительно нужно, например, для выполнения конфигурации сервера, для перезапуска служб, для установки ПО. Для временного получения в Linux права пользователя root применяются сопутствующие команды sudo и su, рассмотренные в данном параграфе.

Sudo позволяет клиенту задать практически любую команду с доступами, разрешениями и привилегиями root. Применять ее очень просто:

sudo <команда_которую_следует_выполнить_с_правами_root>

Если необходимо изменить файл /etc/apt/sources.list*: sudo nano /etc/apt/sources.list.*

Если задать аналогичную команду, но без sudo (просто: nano /etc/apt/sources.list), текстовый редактор запустит файл, но сохранить внесенные изменения вы при этом не сможете, поскольку для завершения задачи не хватит полномочий.

Перед самим выполнением команды она обязательно затребует у вас личный пароль:

sudo nano /etc/apt/sources.list

Password:

Нужно ввести пароль, применяемый для входа в систему, но он должен отличаться от пароля root.

Использовать sudo могут не все, а только профили, внесенные в файл /etc/sudoers. Администратору системы позволено редактировать этот файл с помощью visudo.

Далее рассмотрен весь процесс подробно — с момента создания на примере Ubuntu 18.04.

Первым делом добавим пользователя через adduser (рис. 4.12):

adduser den

Команда добавит den и установит его пароль. Узнать права пользователя Linux с этим логином можно будет в любой момент.

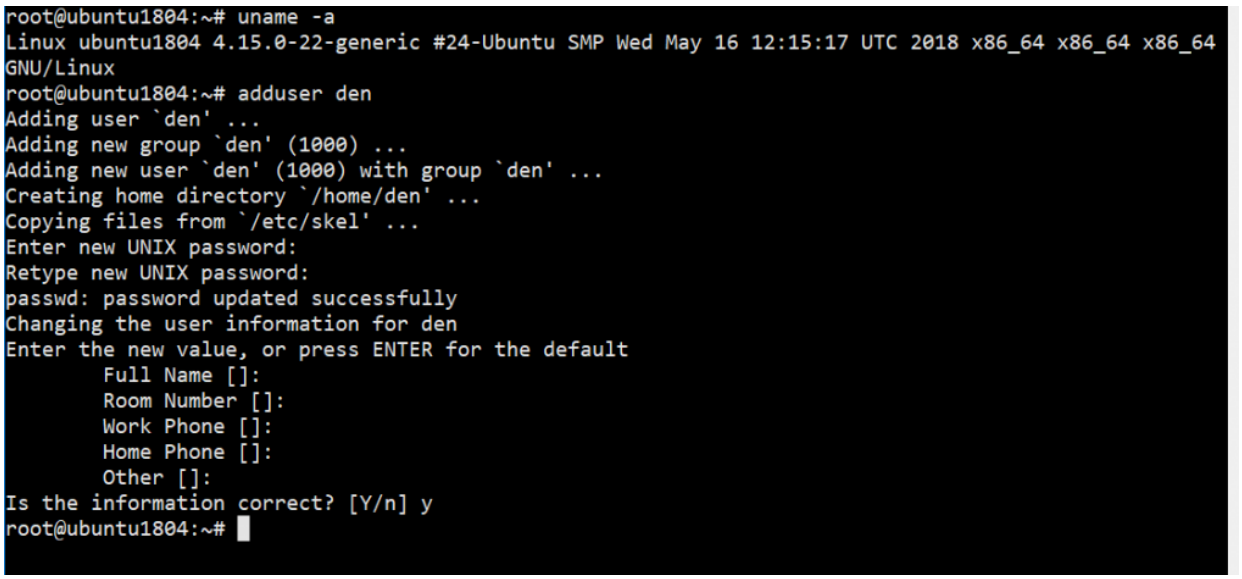

Рисунок 4.12. – Создание пользователя в Ubuntu 18.04.

В Ubuntu adduser не только добавляет в Linux права пользователя, но и создает пароль для него. В прочих дистрибутивах вам понадобится две команды – одна для создания новой пользовательской учетной записи, а вторая — для задания его пароля:

adduser den

passwd den

Затем den нужно добавить в файл /etc/sudoers для редактирования которого предназначена visudo. По умолчанию она использует текстовый редактор vi, который очень неудобен. Гораздо проще установить переменную окружения EDITOR и задать более удобный текстовый редактор, например, nano.

Узнаем, где находится текстовый редактор nano:

which nano

/bin/nano

Установим переменную окружения:

export EDITOR=/bin/nano

```
root@ubuntu1804:~# which nano
/bin/nano
root@ubuntu1804:~# export EDITOR=/bin/nano
root@ubuntu1804:~#
```

Рисунок 4.13. Установка переменной окружения EDITOR.

После этого можно выполнить visudo и дать права пользователю Linux, отредактировав необходимый файл, добавив в него den.

den ALL=(ALL:ALL) ALL

Данная строка объясняет, что den можно выполнять со всех (первый ALL) терминалов, работая как любой (второй ALL) все команды (третий ALL) с максимальными правами. Конечно, его можно и более ограничить, например:

den ALL= /sbin/poweroff

Эта запись означает, что вышло дать права пользователю Linux den и он может с любого терминала вводить команду завершения работы.

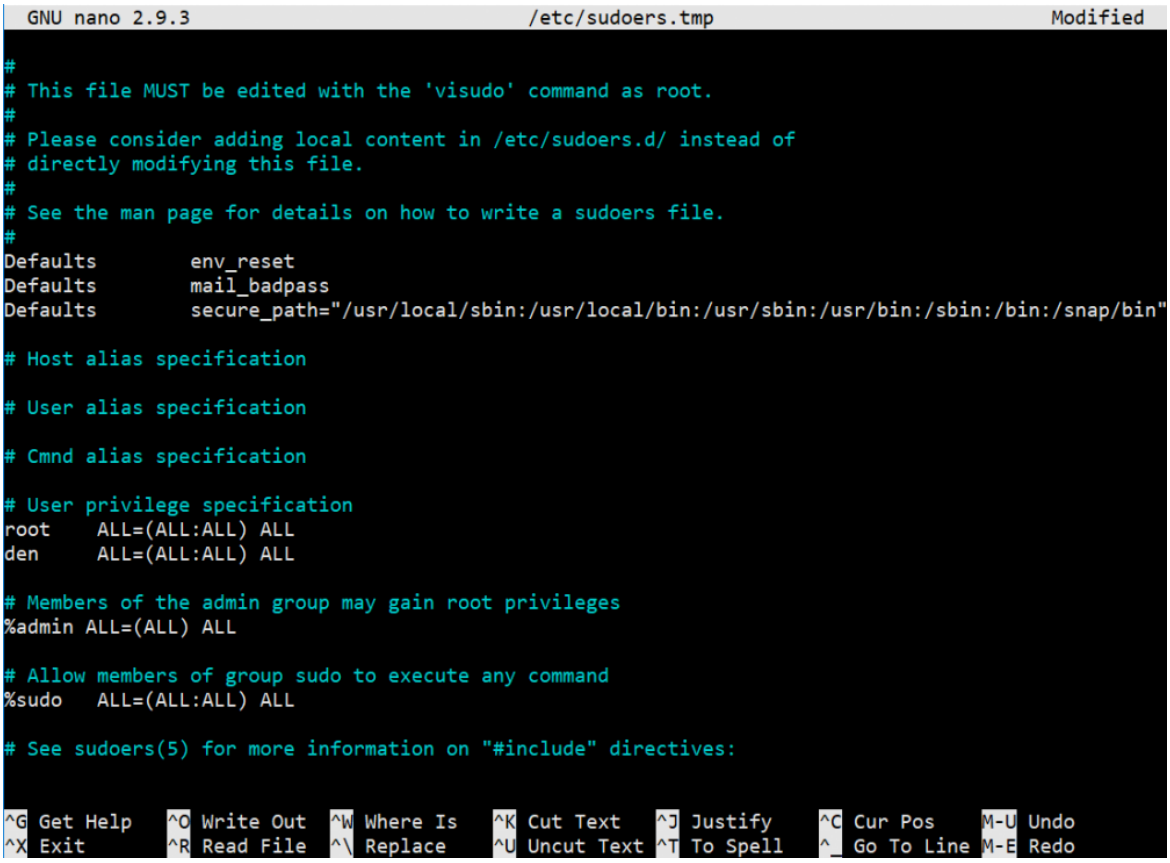

Рисунок 4.14. – Редактирование /etc/sudoers.

Для сохранения изменений нажмите сначала Ctrl+O, а затем Ctrl+X, чтобы выйти из редактора.

Теперь проверим, все ли у нас получилось. Превратимся в den:

su den

Попытаемся с максимальными правами выполнить любую команду, например, ls:

sudo ls

Команда sudo запросит пароль den. Если бы мы допустили ошибку при настройке, sudo сообщила бы, что он не имеет права ее использовать. А поскольку она запросила пароль, мы все сделали правильно.

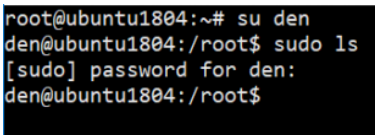

Рисунок 4.15. – Работа sudo.

Каждую команду, которая требует максимальных прав, вам придется выполнять через sudo. Порой нужно выполнить продолжительную настройку и вводить каждый раз sudo не хочется. В этом случае можно запустить командный интерпретатор bash через sudo и вы получите полноценную консоль root. Вот только не забудьте по окончанию настройки ввести exit:

sudo bash

выполняем настройку

exit

Через sudo у вас не получится использовать перенаправление задач категории ввода/вывода, то есть команда вида sudo ls /etc > /root/somefile не сработает. Для этого требуется ввести sudo bash и использовать систему как обычно.

Каждая команда, которая была введена в систему непосредственно с помощью sudo, фиксируется в учетном журнале, находящемся по адресу /var/log/auth.log, и у вас сохранится история введенных задач с полномочиями root, а вот при работе под администратором root журнал вовсе не ведется. Кроме того, если что-то пойдет не так, можно будет понять, что случилось, изучив данный журнал. На рисунке 4.16 видно, что den вводил ls через sudo.

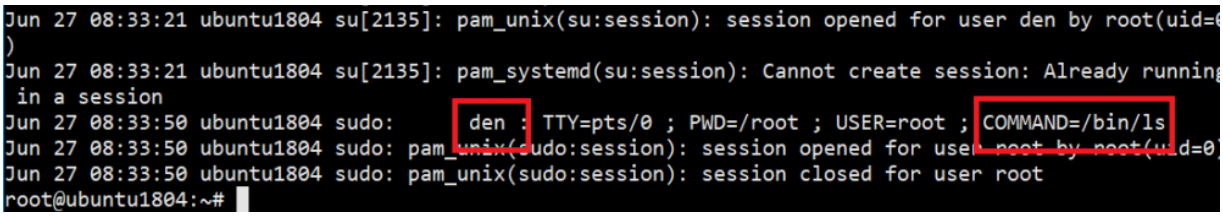

Рисунок 4.16. – Журнал auth.log.

Su дает доступ к консоли с полными правами администратора root любому (даже если он изначально не был вписан в файл /etc/sudoers). Единственное условие — он должен иметь пароль root. Понятно, что в большинстве случаев ним будет сам root, — не станете же вы всем доверять свой пароль? Поэтому su предназначена, в первую очередь, для администратора системы, в Linux права пользователя у него не ограничены, а sudo — для остальных, им тоже иногда требуются права root (чтобы они меньше отвлекали администратора от своей работы).

Использовать su просто:

su

После этого нужно будет подтвердить права root пользователя Linux и ввести пароль, далее вы сможете работать в консоли, как обычно. Использовать su удобнее, чем sudo, потому что нет необходимости вводить su непосредственно перед каждой командой, требующей выполнения с правами root.

Чтобы закрыть сессию su нужно или ввести слово exit, или закрыть окно терминала.

С помощью su можно работать от имени любого пользователя. Ранее было показано, как мы с помощью su den смогли выполнять команды от имени пользователя den.

4.3.5. Управление журналами в Linux (logrotate)

Журналы в Linux иногда разрастаются до неприличных размеров, что приводит к снижению производительности и в некоторых особо запущенных случаях может даже вызвать переполнение дискового пространства. К счастью, администратор может установить себе в помощь утилиту logrotate — настройка Linux для управления журналами.

logrotate — это утилита, выполняющая ротацию и сжатие протоколов. При правильной настройке файл журнала никогда не разрастется до огромных размеров. Если же она не установлена или же неправильно настроена, файловые данные некоторых сервисов (например, Apache) могут занять на диске кучу свободного места.

По умолчанию утилита Logrotate уже предустановлена в Ubuntu 16.04. Она настроена для своевременной обработки журналов всех, имеющихся в системе и используемых пользователями приложений, в том числе rsyslog.

Несмотря на то, что утилита изначально установлена по умолчанию, будет нелишним проверить ее конфигурацию. На одном из моих серверов почему-то не выполнялась ротация Apache, в результате произошло переполнение дискового внутреннего объема.

Logrotate настройка имеет конфигурацию, хранящуюся по следующему адресу:

/etc/logrotate.conf — основной файл. Как правило, он содержит некоторые параметры, предустановленные автоматически, настройки для ряда базовых журналов, не принадлежащим системным пакетам и инструкцию include для подключения конфигурации, хранящейся по адресу /etc/logrotate.d.

/etc/logrotate.d содержит файлы с конфигурацией. Здесь logrotate настройка содержит конфигурацию для rsyslog, apport, dpkg и других системных пакетов. Каждый из них — конфигурация ротации. Вы сможете добавить свои файлы конфигурации в Linux журнал событий, они также будут обработаны утилитой.

Рассмотрим конфигурацию, предлагаемую утилитой logrotate Linux с целью ротации для apport — /etc/logrotate.d/apport:

/var/log/apport.log {
daily
rotate 7
delaycompress
compress
notifempty
missingok
}

Где:

daily — ротацию выполнять однажды в сутки (для редко используемых сервисов можно использовать команду monthly — раз каждый месяц или weekly — раз каждую неделю), удобно при необходимости проведения регулярного аудита.

rotate 7 — позволяет в автоматическом режиме хранить последние 7 журналов.

compress — используется для сжатия обновленного объема данных (стандартно используется технология gzip)

notifempty — запрещает ротировать файл, если он пустой.

missingok — не оставляет сообщение об ошибке в работе системы, если системные журналы Linux пусты.

delaycompress — позволяет на время отсрочить сжатие прошлого журнала до прохождения последующего сдвига по циклу. Эта директива успешно используется лишь в сочетании с compress. Это необходимо, если запущенной

программе невозможно приказать сразу закрыть журнал и допускается определенный период времени продолжать вести запись в используемый для этого ранее файл.

О дополнительных параметрах конфигурации можено узнать в справке (команда man logrotate).

Теперь представим, что у нас есть некий сервис daemond, хранящий свои файлы в каталоге /var/log/daemond. Нужно настроить ротацию этого сервиса.

Все достаточно просто: нужно в /etc/logrotate.d создать daemond (название бывает и иным, важно, понимать, что находится в этом файле сразу по его названию) и заполнить конфигурацию ротации.

Воссоздадим конфигурацию ротации:

touch /etc/logrotate.d/daemond

Конфигурация, установленная пользователем, может быть такой:

/var/log/daemond/.log {*
daily
missingok
rotate 7
compress
notifempty
create 0640 daemon-data daemon-data
sharedscripts
postrotate
systemctl reload daemond
endscript
}

Первая строчка задает требуемые файлы. Здесь это все лог-файлы из используемого каталога /var/log/daemond. Команда create определяет, что после ротации будет создан новый пустой файл с правами 0640, владельцем daemon-data и группой daemon-data. Если ваш сервис выполняется от имени какого-то

пользователя (например, daemon-data), а не от root, то очень важно указать правильно имя пользователя и задать название группы.

Параметр sharedscripts говорит о том, что скрипт под названием postrotate будет выполняться только лишь 1 раз, а не непосредственно после детальной обработки любого имеющегося файла.

Параметры postrotate и endscript позволяют указать скрипт, который будет запущен только уже после того, как журнал полностью обновится. В указанном примере приложение перезагружается.

Следующим шагом является настройка утилиты на автоматический запуск. Чтобы она запускалась автоматически, введите команду crontab -e и в появившемся редакторе добавьте строку:

*10 * * * * /usr/sbin/logrotate /etc/logrotate.conf*

Сохраните файл и выполните выход из редактора. Мы создали расписание планировщика, которое будет выполнять logrotate на 10-ой минуте каждого часа. Конфигурация будет загружаться из /etc/logrotate.conf.

5. АДМИНИСТРИРОВАНИЕ ЭЛЕМЕНТОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ СЕРВЕРОВ НА ОСНОВЕ ОПЕРАЦИОННЫХ СИСТЕМ WINDOWS

5.1. Общие рекомендации по обеспечению безопасности Windows Server

5.1.1. Изоляция серверных ролей

Одна из главных задач планирования безопасности перед развертыванием — сокращение контактной зоны сервера. Принцип сокращения контактной зоны основывается на том предположении, что чем больше кода работает в системе, тем больше вероятность, что в нем найдется уязвимость, которой может воспользоваться злоумышленник. Поэтому для сокращения контактной зоны нужно обеспечить, чтобы на сервера выполнялся только необходимый код.

Однако для максимизации защиты рекомендуется зайти немного дальше. В общем случае рекомендуется конфигурировать каждый сервер на выполнение одной конкретной задачи. Например, вместо того чтобы запускать службы DNS и DHCP на машине, уже выполняющей роль файлового сервера, с точки зрения безопасности лучше устанавливать каждую роль на выделенном сервере. Это не только позволяет сократить контактную зону, но также упрощает устранение неполадок, потому что конфигурация серверов значительно проще.

Понятно, что иногда использования отдельных серверов для каждой роли непрактично из соображений экономии или функциональных требований. Но даже в таких обстоятельствах, всегда лучше по возможности изолировать серверные роли.

Дополнительную экономию средств на серверы можно получить за счет виртуализации серверов. Например, лицензия на редакцию Windows Server 2008 R2 Enterprise предусматривает возможность размещения до четырех виртуальных машин при условии, что на сервере установлена только роль Hyper-V.

5.1.2. Форсирование безопасности DNS с помощью DNSSec

Недостатки открытого DNS-севера

Системы разрешения имен DNS, являющаяся одной из основ современной системе сетевых взаимодействий, разрабатывалась более 20 лет назад, когда о вопросах защиты информации почти не задумывались. Одним из основных недостатков системы DNS является возможность подделки ответа на DNS запрос.

Проблема в том, достоверность ответа DNS-сервера никак не проверяется, это означает, что в случае взлома DNS сервера (и перенаправления его на ложные DNS сервера), подделки DNS записи, отравлении кэша DNS (DNS cache poisoning), можно отправить пользователя на произвольный IP адрес, причем пользователь будет находится в полной уверенности, что он работает с легитимным сайтом или сервисом. Этими методиками широко используют злоумышленники, перенаправляя пользователей на сайты, содержащие вредоносные коды или собирая их личные данные (пароли, номера кредиток и т.п.) с помощью т.н. pharming атак.

Зачем нужна технология DNSSec

DNS Security Extensions (DNSSEC) – технология, предназначена для повышения безопасности службы DNS за счет использования криптографических подписей, позволяющих однозначно удостоверится в подлинности информации, полученной от DNS сервера. Т.е. все запросы и ответы DNS сервера с поддержкой DNSSEC должны обладать цифровой подписью, верность которой может проверить клиент с помощью открытого ключа. Эти цифровые подписи создаются при подписывании зоны (применения к ней DNSSEC).

Упрощенно механизм проверки DNSSEC работает так: клиент отправляет запрос на DNS сервер, сервер возвращает DNS ответ с цифровой подписью. Т.к. клиент обладает открытым ключом центра сертификации, подписавшего записи DNS, он может расшифровать подпись (хэш-значение) и проверить ответ DNS.

Для этого и клиент и сервер DNS должны быть настроены на использование одного якоря доверия (trust anchor).Trust anchor – предварительно настроенный открытый ключ, связанный с конкретной зоной DNS. Если DNS сервер поддерживает несколько зон, то может использоваться несколько якорей доверия.

Важно отметить, что основное предназначение DNSSEC – защита от подделки и модификации DNS-запросов и ответов. Но сами передаваемые по сети данные не шифруются (хотя конфиденциальность передаваемых DNS данных и может быть обеспечено с помощью шифрования, но это опционально и не является основной целью DNSSEC).

Основные компоненты DNSSEC определены в следующих стандартах RFC:

RFC 4033

RFC 4034

RFC 4035

DNSSEC в системах Windows

Поддержка технология DNSSec появилась еще в Windows Server 2008 R2 и Windows 7. Однако из-за сложности и неочевидности настроек, широкого распространения она не получила. Свое дальнейшее развитие DNSSec получила в Windows Server 2012, в котором функционал DNSSec был существенно расширен, и предполагает возможность настройки через графический интерфейс.

5.1.3. Изменение стандартный порт RDP-сессий

По умолчанию RDP слушает tcp порт 3389, брутфорсеры и прочие злоумышленники постоянно сканируют сеть на наличие открытого порта 3389 и 445, подбирая пароли, кормят протокол эксплоитами, сервер из-за этого в лучшем случае может зависнуть или уйти в BSOD, а в худшем он может быть скомпрометирован. Поэтому стоит изменить стандартный порт 3389 на любой

другой, свободный порт. Сделать это можно специальной утилитой от microsoft или изменив ключ реестра.

Перед изменением порта, необходимо добавить соответствующее правило в брандмауэр и записать где-нибудь порт, но не на том сервере, на котором производилась замена.

5.1.4. Конфигурируем брендмауэр: запрещаем всё кроме нужного

Редактируем правила входящих подключений в брандмауэре, отключаем всё что не нужно, например, обычно на веб-сервере не нужно ничего кроме:

- DNS;
- HTTP;
- HTTPS;
- FTP;
- RDP.

5.1.5. Защита от намеренного подбора паролей к учетным записям

После проведения вышеописанных действий, а именно по изменению порта, то теперь следует настроить защиту от намеренного подбора паролей, делается это следующим образом:

Переходим в локальную политику Control Panel\System and Security\Administrative Tools\Local Security Policy далее в Account Policies — Account Lockout Police.

Account lockout threshold — тут задается количество неудачных попыток входа до блокировки учетной записи (минимально необходимое количество 5 раз).

Account lockout duration — устанавливается время, через которое учетная запись будет автоматически разблокирована (необходимо установить не более 20 минут).

Reset account lockout counter after — сброс счетчика неудачных попыток входа (необходимо установить не более 20 минут).

5.1.6. Изменение имени локального администратора

Для выполнения данных действий необходимо зайти в локальную политику Control Panel\System and Security\Administrative Tools\Local Security Policy, перейти в Local Policies — Security Options — Account: Rename administrator account, открыть правило и ввести новое имя учетной записи администратора.

Примечание: выше есть правило отключения аккаунта администратора (Accounts: Administrator account status), но лучше просто переименовать. Дело в том, что учетные записи пользователей блокируются после определенного количества неудачных попыток входа в систему, и только аккаунт администратора не подвержен блокировкам, сделано это специально, чтобы администратор не потерял контроль над сервером в период блокировки всех остальных учетных записей.

5.1.7. Отключение показа пользователей сервера при интерактивном подключении

Переходим в Local Policies — Security Options и меняем следующие привила:

Interactive logon: Display user information when the session is locket

Открываем правило и в выпадающем меню выбираем «Do not display user information»

Interactive logon: Do not display last user name

Открываем правило и выставляем «Enabled»

5.1.8. Patch-менеджмент (политика обновлений)

Как только вы запустили новый сервер, нужно позаботится о том, чтобы до ввода в эксплуатацию, были установлены все доступные обновления. В процессе работы тоже очень желательно устанавливать выходящие обновления, для этого можно выделять 1 день в месяц, непосредственная недоступность сервера обычно занимает в районе 15 минут.

Динамическое управление доступом

Многие годы в Windows используется механизм управления доступом на уровне пользователей и групп. Пользователь может работать в защищенной локальной сети или подключаться через общественную точку доступа. Человек один, а риски разные.

Особо остро этот вопрос стоит сегодня, так как по статистике большинство утечек происходит по вине инсайдеров (намеренной или случайной), которые имеют легальный доступ к некоторой информации. В результате, чтобы перекрыть все потребности, создается большое количество групп, что серьезно усложняет администрирование, в частности понимание того, кто и куда действительно имеет доступ. Малейшая ошибка пользователя или админа — и документ оказывается не на своем месте и имеет ненадлежащие права доступа. Современным организациям остро необходим простой в использовании механизм предотвращения утечки информации (DLP, Data Leak Prevention).

Защитить содержимое можно при помощи службы управления правами (Rights Management Services), но она снимает только часть проблем. Более глобально задачу управления доступом и аудита призвана решить технология динамического управления доступом (Dynamic Access Controls, DAC).

Технология базируется на трех основных понятиях:

– *классификация документов* — на основе тегов, которые добавляются пользователем при создании/редактировании документа (в свойствах), приложением, наследуются от каталога или присваиваются по контексту. Если

документ не классифицирован, то используются только традиционные средства доступа;

– *политики* — состоят из одного или нескольких правил, основанных на выражениях, описывающих условия доступа для утверждений пользователей/устройств и тегов. Выражения содержат атрибуты Active Directory и, по сути, являются основой DAC, показывая, кто и на каких условиях может получить доступ;

– *аудит* — расширенные политики аудита, позволяющие получить информацию о попытках доступа к конфиденциальной информации.

Реализована интеграция DAC со службой RMS, что позволяет в реальном времени защищать документы, которым присвоен соответствующий тег. Настройки упрощает автоматическое тегирование документов, которое создается при помощи правил, настраиваемых в диспетчере ресурсов файлового сервера (File Server Resource Manager).

Для реализации DAC система безопасности Win2012 получила новый механизм утверждения claims, такие утверждения существуют для ресурсов, пользователей и учетных записей компьютеров. При входе в систему помимо идентификатора безопасности SID (Security Identifier) учетной записи в маркер добавляются claims (просмотреть их можно, введя «whoami /claims»). Вот эти утверждения, вместе с данными об участии в группах, и используются при доступе к объектам файловой системы. При этом старые механизмы никуда не исчезли. Вначале применяются права доступа к сетевому ресурсу, затем NTFS и, наконец, вступает в работу DAC.

Использование технологию доступа к сети NAP

Microsoft разработала технологию NAP (Network Access Protection), которая представлена в серверной операционной системе Windows Server 2008. Данная технология предназначена для блокирования/постановки в карантин компьютеров, не соответствующих политикам, принятым в локальной сети. Таким образом обеспечивается защита локальной сети.

Технология NAP предлагает различные варианты реализации: выдача IP-адресов из различных диапазонов для «хороших» и «плохих» компьютеров; установление соединения IPSec с «хорошими» компьютерами, в то время как «плохим» компьютерам в соединении будет отказано.

С помощью NetWork Access Protection администраторы компании могут поддерживать состояние «здоровья» сети. Параметры системы клиента проверяются на соответствие политике безопасности, например, наличие свежих обновлений операционной системы, наличие антивирусной программы и состояние её обновлений, установлен и работает ли на клиенте сетевой экран. На основе этих параметров каждый компьютер получает свою оценку безопасности. Компьютер, удовлетворяющий требованиям системы контроля, получает доступ в сеть предприятия. Компьютеры, не удовлетворяющие требованиям безопасности, не смогут получить доступа в сеть или смогут получить доступ лишь в изолированную часть сети, предоставляющую сервисы для достижения клиентом требуемого уровня безопасности.

5.2. Настройка терминального сервера в облаке

Терминальные серверы — не новинка и существуют еще со времен мейнфреймов. Ранее терминальные серверы использовались по причине ограниченных системных ресурсов клиентов. Сейчас же они используются для большего удобства администрирования — настраиваете терминальный сервер, устанавливаете приложение (например, 1С) и предоставляете всем пользователям, которым необходим доступ к этому приложению. Преимущество в том, что пользователи могут находиться в разных городах и даже странах и они не ограничены только корпоративной локальной сетью.

В наше время вся IT-инфраструктура постепенно «переезжает» в облака, поскольку так проще, удобнее и даже дешевле.

Терминальный сервер – популярное решение, позволяющее создать для организации единую ИТ-среду. Его используют для того, чтобы доставить

приложения на машины, тонкие клиенты и мобильные устройства. В этом случае пользователь может и не догадываться о том, что вся работа программы происходит на сервере, а ПО даже не установлено на компьютер.

Выбор терминального сервера – важный процесс, который должен происходит с учетом ряда требований и критериев. Разберемся, на что обратить внимание в первую очередь и как выбрать подходящую конфигурацию для бизнес-процессов.

Для начала стоит понять, какие задачи можно решить при помощи такой технологии:

– повышение эффективности работы программ и приложений за счет «близости данных». То есть приложения устанавливаются прямо в ЦОД и потом перераспределяются на требуемые рабочие места.

– перевод сотрудников на тонкие клиенты. Этот процесс подразумевает замену стандартных компьютеров на миниатюрные устройства, подключаемые к сервису. Благодаря этому происходит оптимизация технической поддержки на местах.

– унификация рабочих мест. Возможно загрузить одинаковое ПО на всех рабочих местах, не потребуется делать это для каждого компьютера.

Но добиться этого удастся только при правильном подборе сервиса. Важно понимать, что не существует универсального решения для всех компаний.

Память терминального сервера будет использоваться точно также, как и на обычном компьютере. Но важно понимать, что делать это будут все пользователи одновременно. То есть если для работы программы требуется 20 мб, то для десяти пользователей потребуется 200 мб и т. д.

Лучше всего выбирать оперативную память не менее 512 мб, а для крупных компаний показатель будет в разы больше. Предварительно посчитайте, какой минимальный показатель подойдет для вас, и только после этого приступайте к выбору сервера терминалов.

Лучше всего, если на жестких дисках будет храниться только операционная система, приложения и требуемые файлы. Также потребуется свободное место для временных файлов и профилей.

Остальные данные храните отдельно от жестких дисков. Во-первых, это поможет повысить безопасность файлов и избежать их утраты. Во-вторых, вы обеспечите бесперебойную работу терминала, а, следовательно, не потребуется увеличивать объем дисковых хранилищ.

В идеале используйте идентичные серверы, чтобы пользователи могли разделять нагрузку между ними и при необходимости можно было заменить отдельный сервис.

При подборе необходимого процессора можно не тратить лишнее время на подсчеты параметров. Дело в том, что скорость и производительность процессоров диктуется производителями оборудования, то есть «не нужно изобретать велосипед» – выбирайте из того, что есть.

Идеальное решение – использование такого количества процессоров, которые соответствуют число процессов в операционной системе. Однако добиться такого практически нереально, поэтому выбирайте 1, 2, 4 или 8 ядерный процессор.

Помните о том, что в терминальном сервере используется алгоритм кооперативной многозадачности. Это означает, что каждый сеанс одного пользователя выполняет запросы только по мере надобности. То есть если сотруднику в данный момент не требуется выполнять какие-то сложные задачи, то другим работникам достанется больше вычислительных мощностей.

Чаще всего загрузка процессора будет постоянно на уровне 98-99%. Это абсолютно нормальные показатели, так как они означают, что параметры оборудования были подобраны с максимальной точностью.

При выборе терминального сервера обязательно потребуется подобрать тип его организации. Возможно два варианта:

– ***RDS***. Сотрудники смогут работать с одной копией ОС, подключая к ней несколько сессий одновременно. Однако такое решение достаточно уязвимое,

так как сбой в одной из сессий приведет к проблемам в остальных. Лучше использовать RDS для небольшого количества пользователей.

– ***VDI***. В этом случае каждый работник использует изолированную версию операционной системы. То есть, на каждого пользователя приходится определенное количество ресурсов.

Для выбора стоит учесть несколько факторов. С одной стороны, это общее количество сотрудников, которые будут работать с терминалом. С другой, количество ресурсов, необходимое для каждого пользователя.

После выбора терминального сервера стоит протестировать его на производительность. Вполне возможно, что вычислительных мощностей и объема памяти окажется недостаточным для текущего количества пользователей.

Тестирование системы должно помочь ответить на следующие вопросы:

Максимальное количество пользователей, которые смогут работать на терминале, загружая его до 99%?

Как поведет себя сервер при максимальных нагрузках и будут ли возникать сбои в его работе?

Для правильного тестирования системы выберите приложение для проверки. Лучше всего выбирать программы, наиболее важные для вашего бизнеса. Далее выполните следующие шаги:

– определите тестовые задания для проверки;

– обозначьте требуемое время ответа сервера;

– посмотрите, как пользователи работают с приложениями;

– создайте сценарий для моделирования работы приложений в разных ситуациях;

– проанализируйте полученные результаты и найдите «узкие места» в системе.

Полученные данные помогут понять, необходим ли пересмотр параметров терминального сервера и какие проблемы в его работе могут возникнуть.

Помните, что терминал должен способствовать эффективной работе сотрудников, а не мешать им решать требуемые бизнес-задачи.

5.2.1. Создание виртуального сервера

Для того чтобы терминальный сервер настроить первым делом нужно создать виртуальный сервер. Для создания сервера терминалов необходимо собрать соответствующую конфигурацию. Двумя ядрами уже не обойдешься, как минимум четыре ядра, 12 Гб оперативной памяти и SAS-диск на 120 Гб.

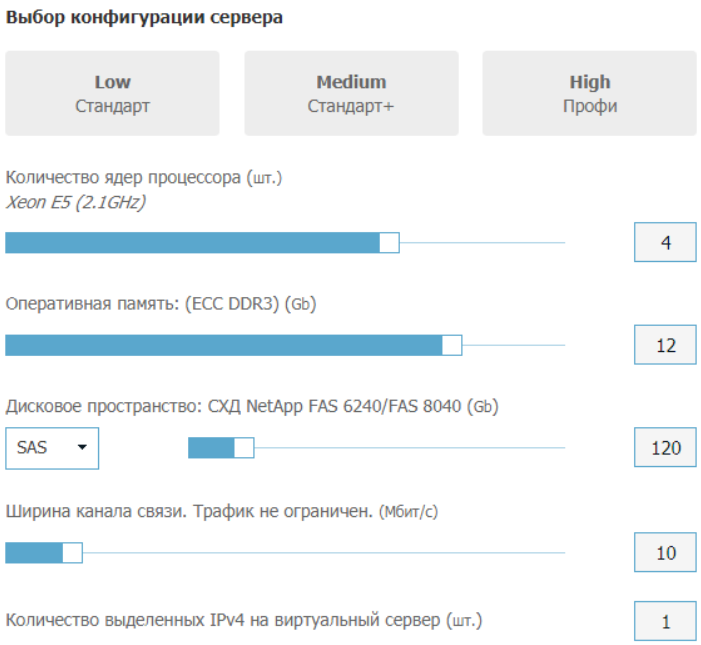

Рисунок 5.1. Минимальная конфигурация сервера.

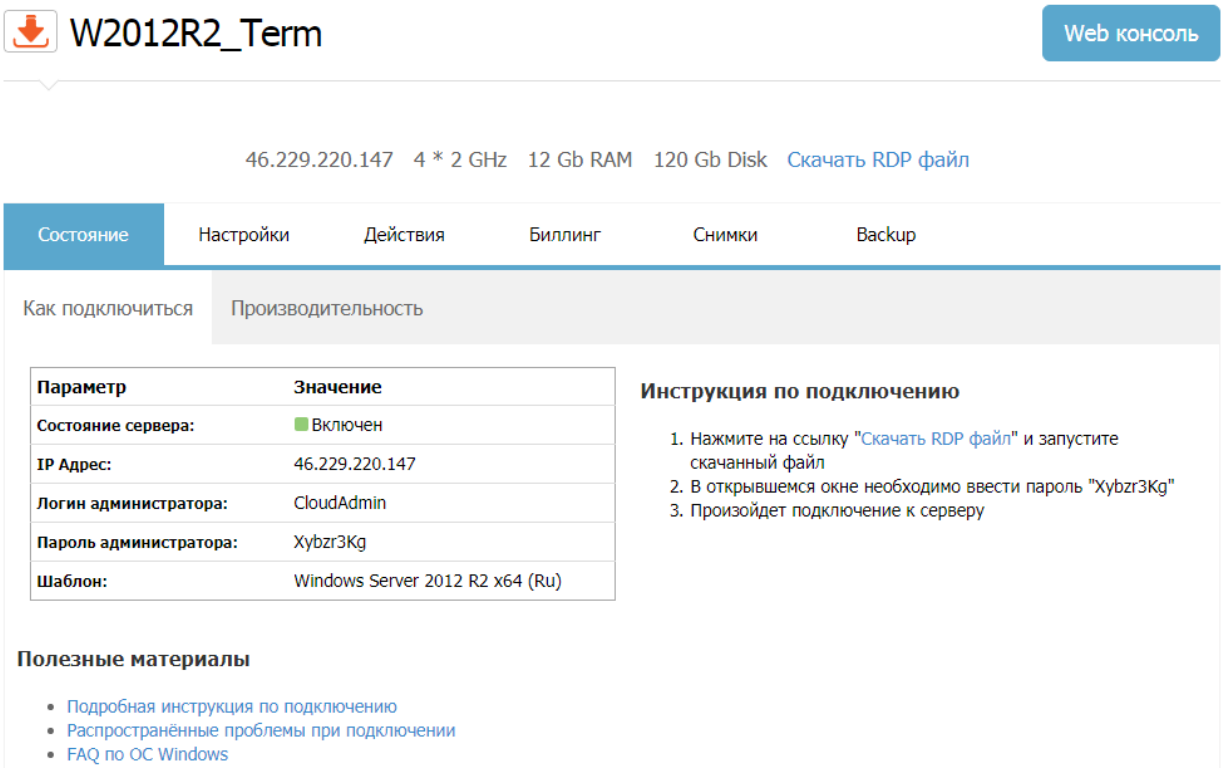

Рисунок 5.2. Созданный виртуальный сервер.

Подключение к серверу осуществляется с использованием данных, предоставленных в панели управления.

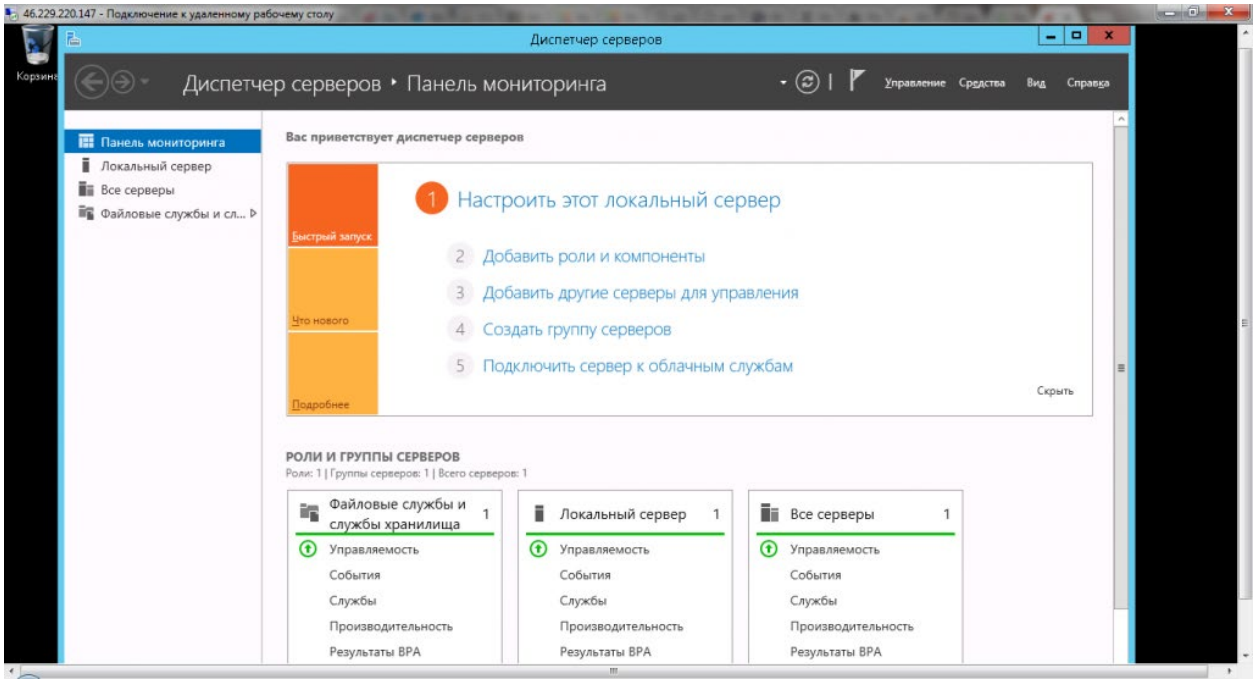

Рисунок 5.3. Удаленное соединение с сервером.

Следующим шагом будет выступать установка службы удаленных рабочих столов. Для этого необходимо запустить диспетчер серверов выполнением команды servermanager.exe или при помощи ярлыка на панели задач. В меню управление необходимо выбрать команду «Добавить роли и компоненты» как указано на рисунке 5.4, далее откроется «Мастер добавления ролей и компонентов» как указано на рисунке 5.5.

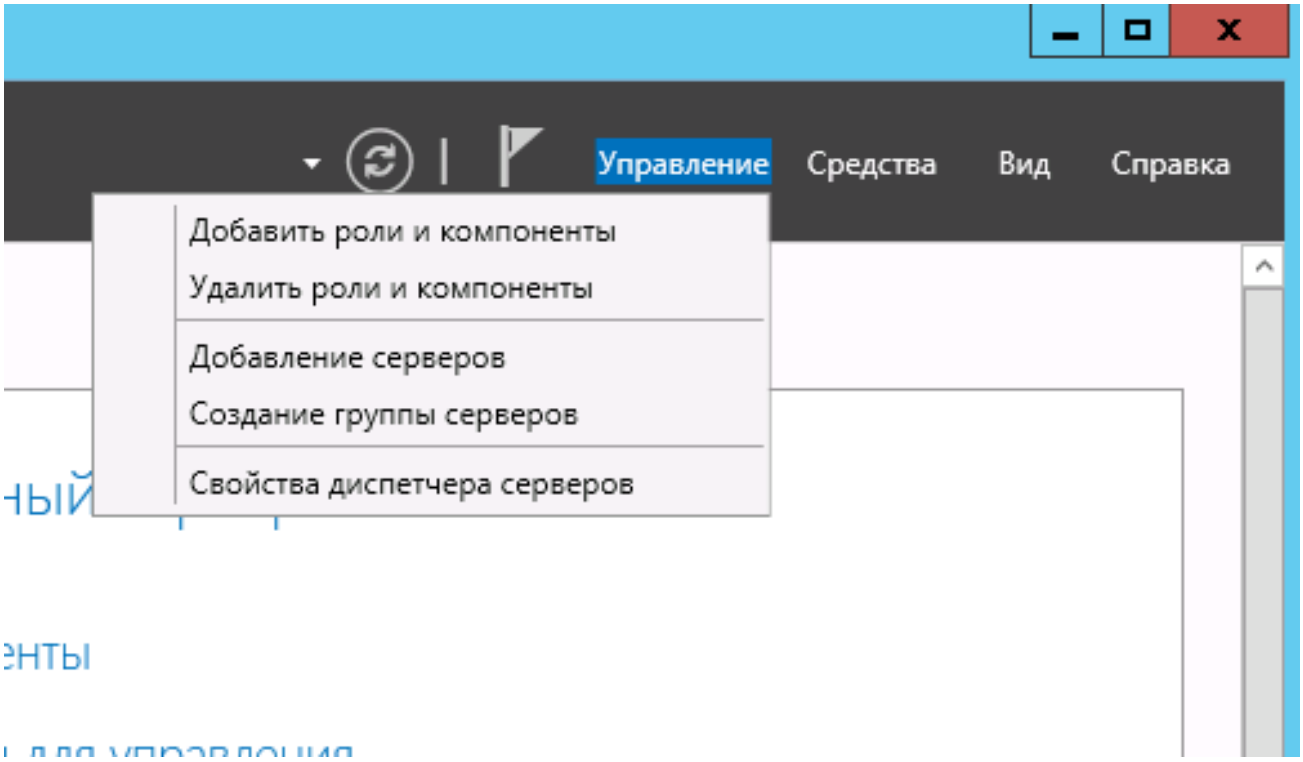

Рисунок 5.4. Меню управление.

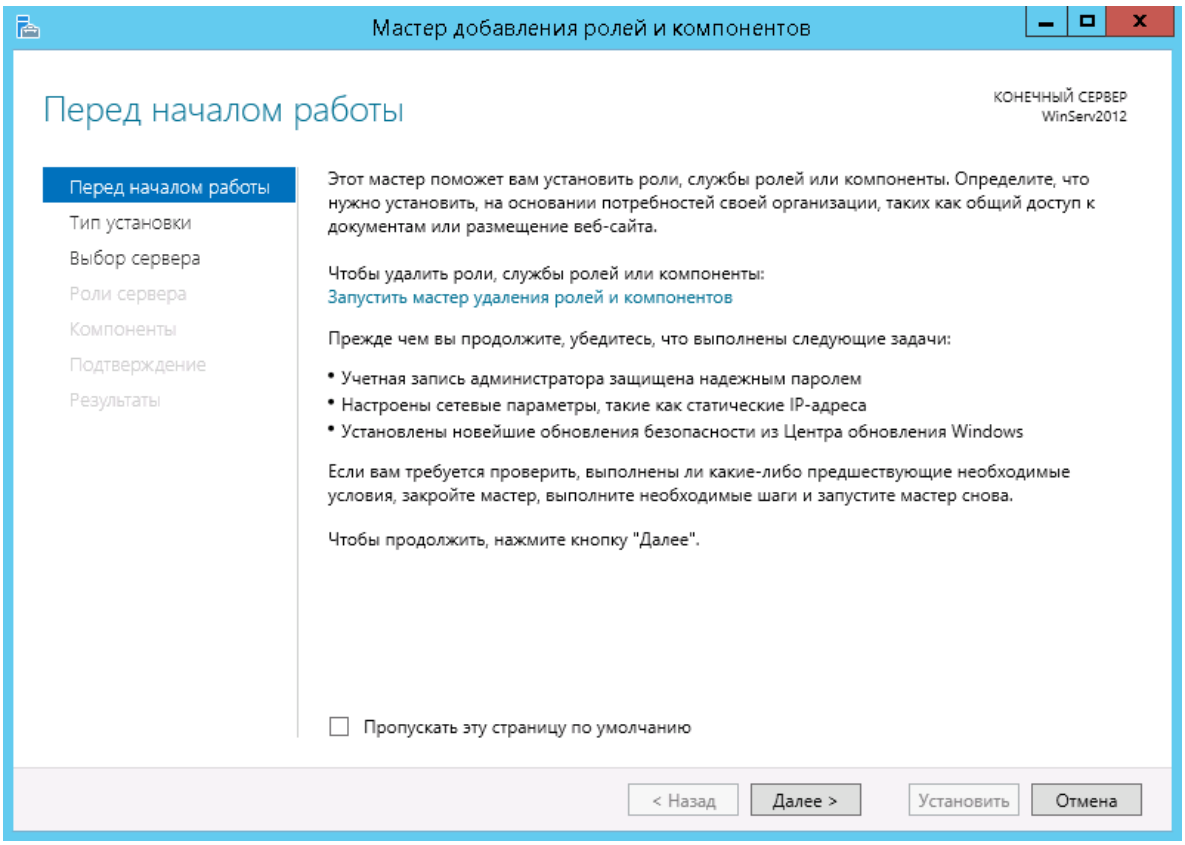

Рисунок 5.5. Мастер добавления ролей и компонентов.

Далее необходимо выполнить следующие действия:

1. Нажмите кнопку **Далее**.

2. Оставьте переключатель в положении **Установка ролей или компонентов** (рис. 5.6) и снова нажмите кнопку **Далее**.

3. Выберите сервер из пула, на который нужно установить службу терминалов. В нашем случае будет один сервер (рис. 5.7). Нажмите кнопку **Далее**.

4. Отметьте роль **Службы удаленных рабочих столов** и нажмите кнопку **Далее**.

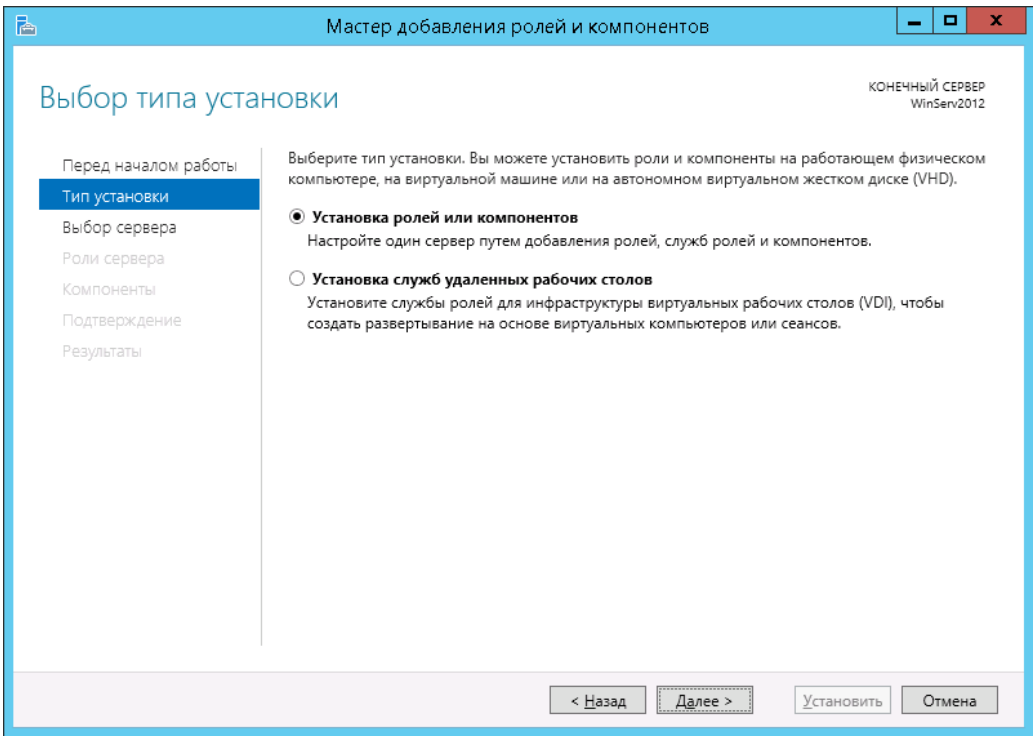

Рисунок 5.6. Мастер добавления ролей и компонентов.

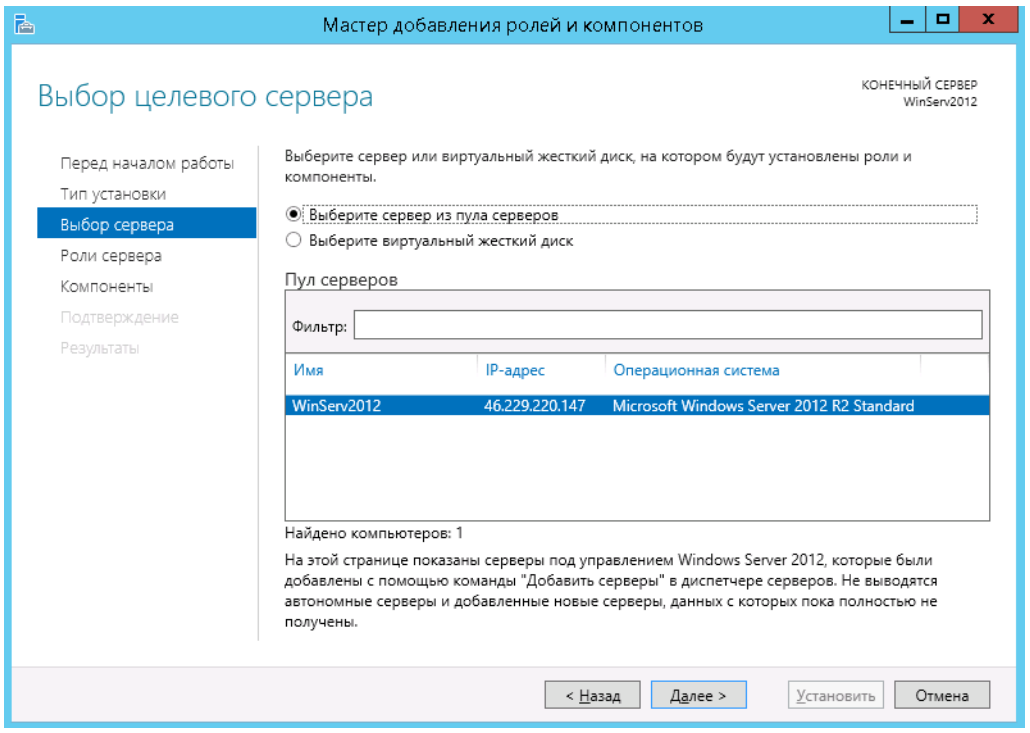

Рисунок 5.7. Выбор сервера.

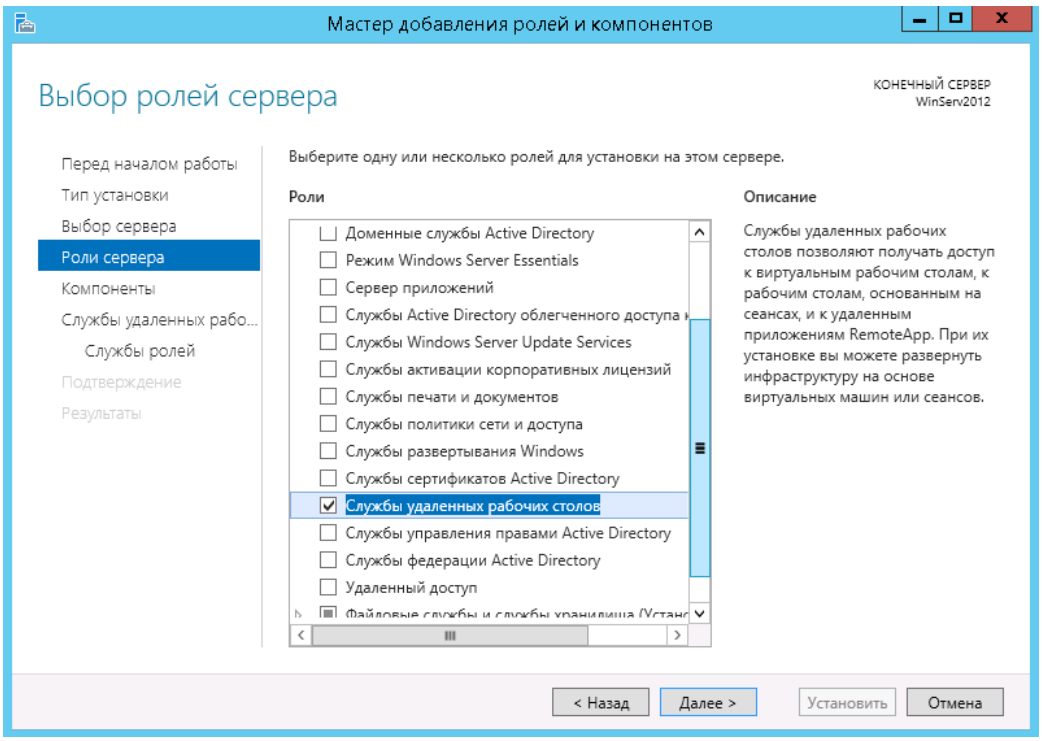

Рисунок 5.8. Выбор службы удаленных рабочих столов.

5. Компоненты оставьте без изменения, то есть на следующем экране **Мастера добавления ролей и компонентов** просто нажмите **Далее**.

6. Мастер отобразит описание роли «Службы удаленных рабочих столов» (рис. 5.9). Нажмите кнопку **Далее**.

7. Выберите устанавливаемые службы ролей. Установите **Лицензирование удаленных рабочих столов**. Сразу после выбора этой службы необходимо согласиться на установку дополнительных компонентов, нажав кнопку **добавить компоненты** (рис. 5.10).

8. Также нам понадобится служба **Узел сеансов удаленных рабочих столов** (рис. 5.11). Как и в прошлом случае, нужно согласиться на добавление дополнительных компонентов.

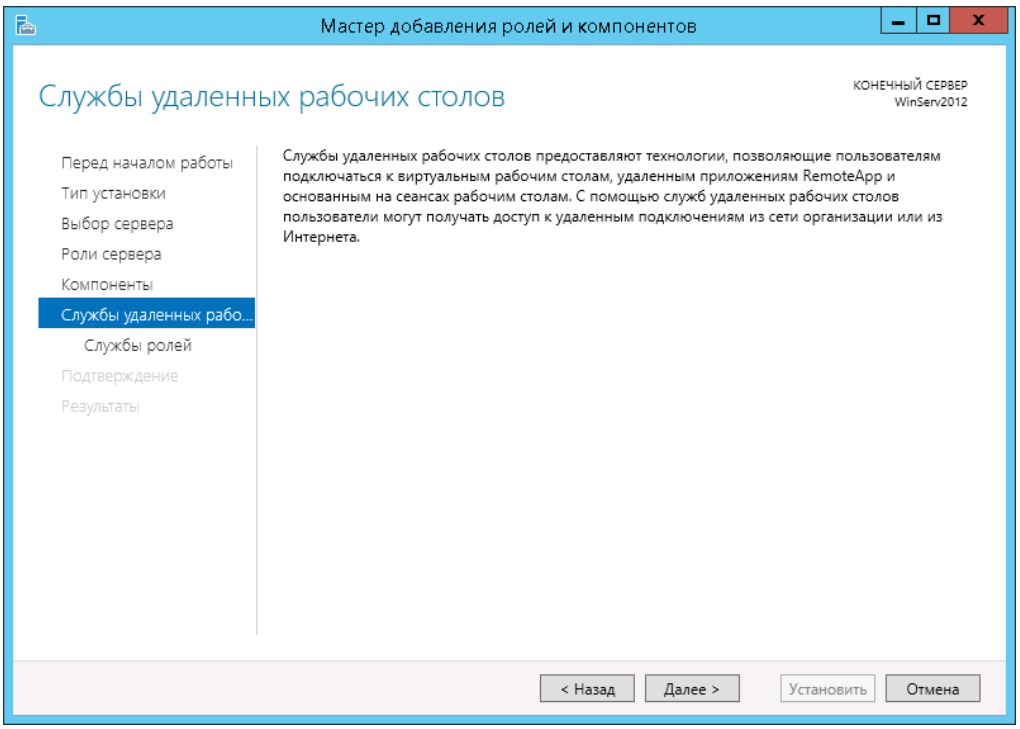

Рисунок 5.9. Службы удаленных рабочих столов.

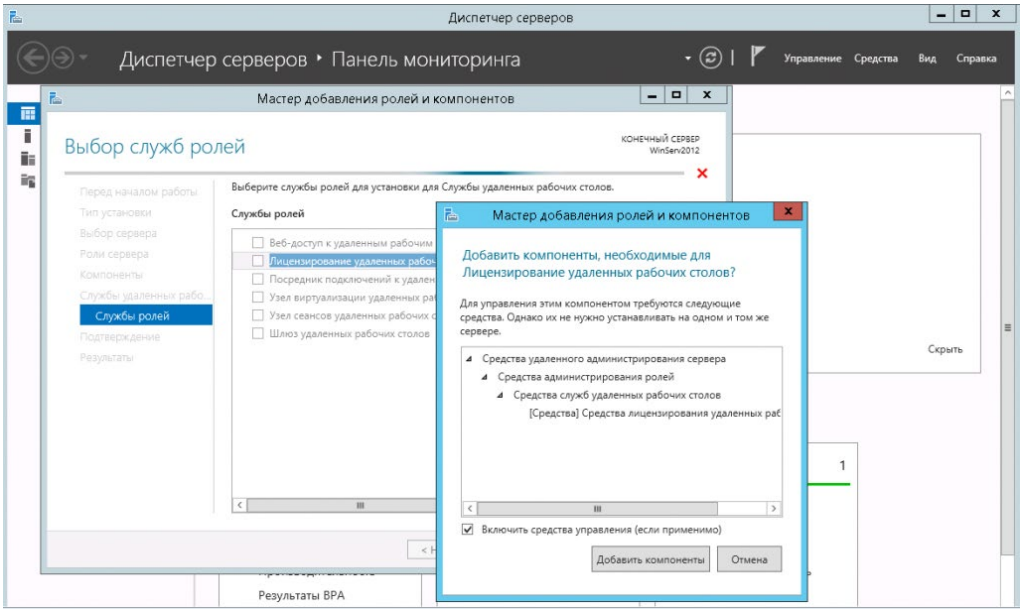

Рисунок 5.10. Лицензирование удаленных рабочих столов.

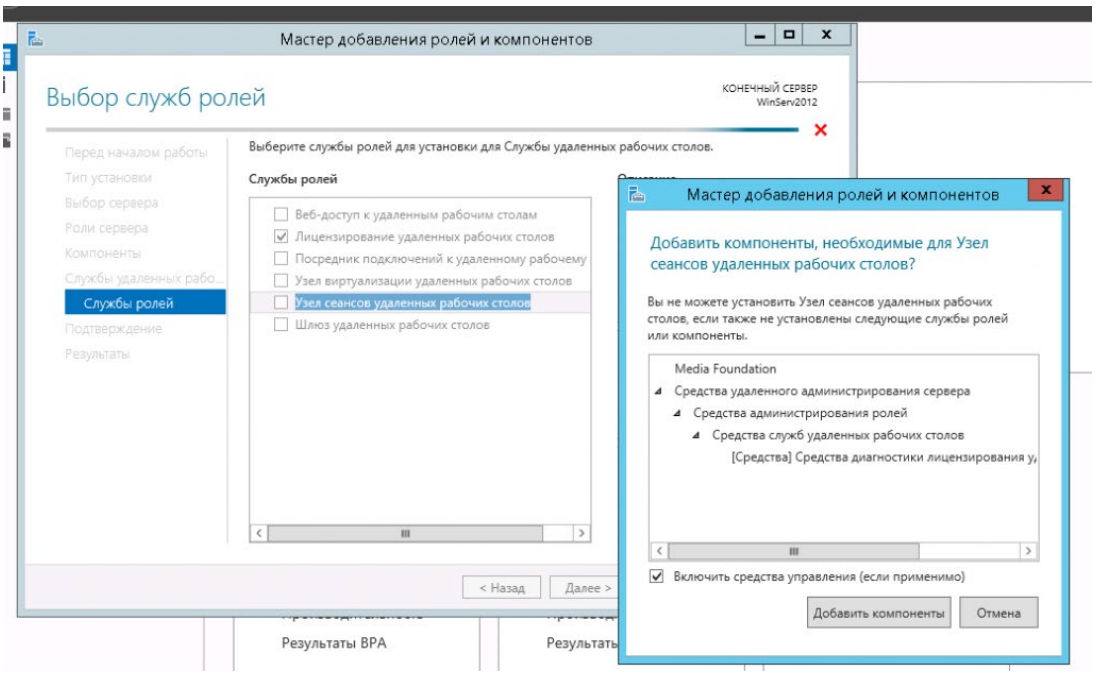

Рисунок 5.11. Установка службы Узел сеансов удаленных рабочих столов.

9. Мы определили все параметры установки роли. На последней странице мастера необходимо включить параметр **Автоматический перезапуск конечного сервера**, если требуется и нажмите кнопку **установить** (рис. 5.12).

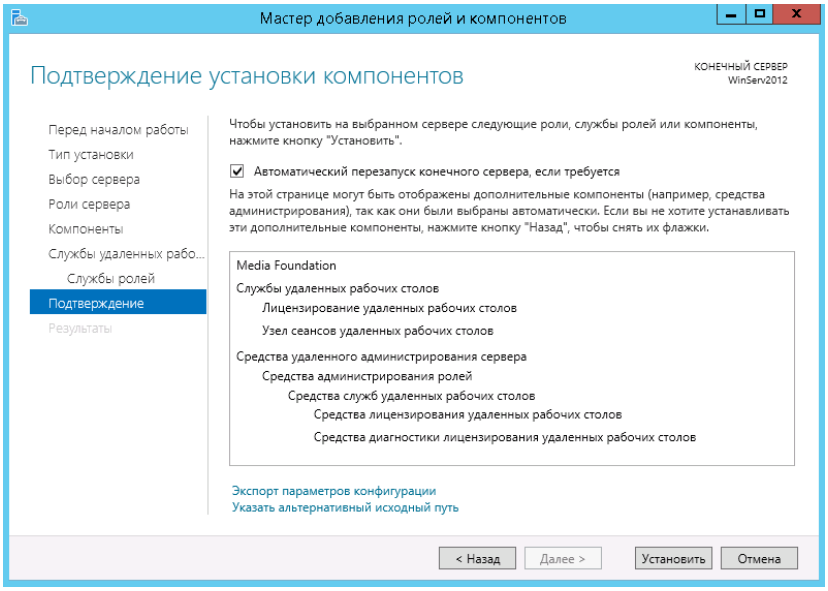

Рисунок 5.12. Подтверждение установки компонентов.

Осталось дождаться установки роли. Если все пройдет хорошо, после перезагрузки вы увидите сообщение об успешной установке всех выбранных служб и компонент. Просто нажмите кнопку **закрыть** для завершения работы мастера.

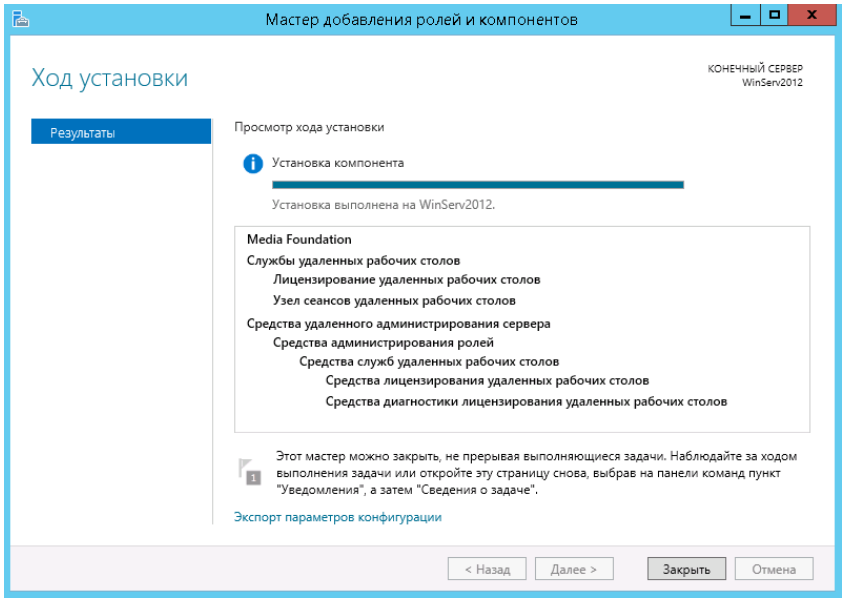

Рисунок 5.13. Установка завершена.

5.2.2. Настройка сервера лицензирования для удаленных рабочих столов

Первым шагом по настройке сервера лицензирования для удаленных рабочих столов будет являться запуск **средства диагностики лицензирования удаленных рабочих столов.** Для этого необходимо выбрать соответствующую команду из меню **средства, Terminal Services** диспетчера серверов.

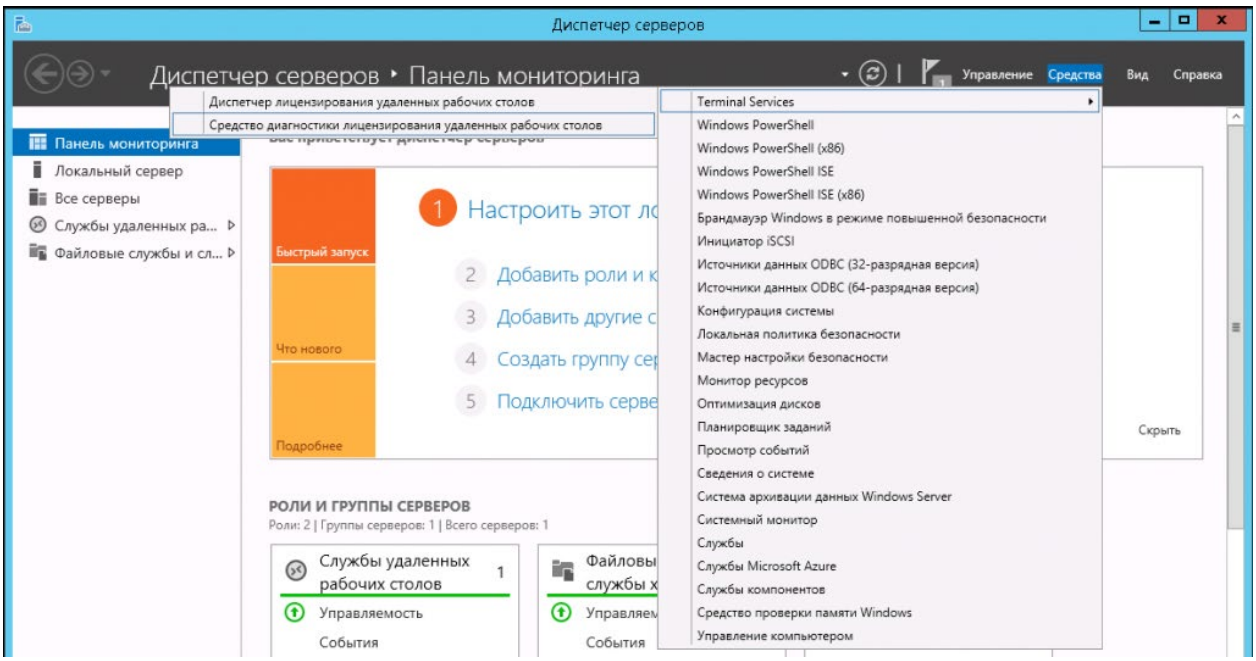

Рисунок 5.14. Запуск средства диагностики лицензирования удаленных рабочих столов.

Средство диагностики сообщит, что доступных лицензий пока нет, поскольку не задан режим лицензирования для сервера узла сеансов удаленных рабочих столов (рис. 5.15). Также средство диагностики сообщит, что льготный период (по умолчанию 120 дней) еще не истек, но этот сервер еще не настроен на использование хотя бы одного сервера лицензирования.

Относительно льготного периода нужно знать следующее:

существует льготный период, в течение которого сервер лицензирования не требуется, однако после его истечения для подключения к серверу клиенты должны использовать действительную клиентскую лицензию служб удаленных рабочих столов, выданную сервером лицензирования;

удаленный рабочий стол поддерживает два одновременных подключения для удаленного администрирования компьютера. Для этих подключений сервер лицензирования не требуется.

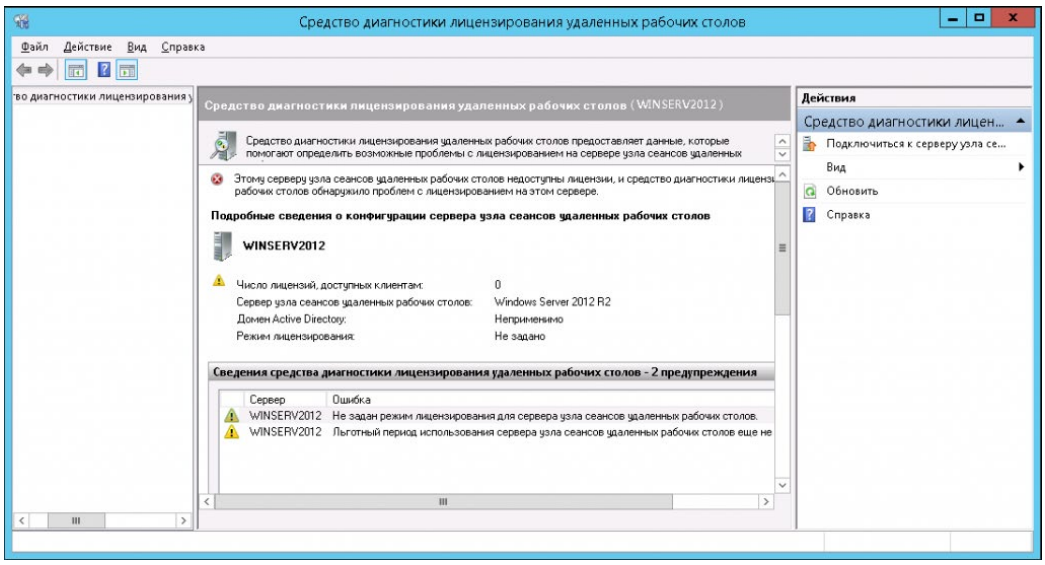

Рисунок 5.15. Запуск средства диагностики лицензирования удаленных рабочих столов.

В Windows Server 2012 сервер лицензирования указывается в локальных групповых политиках, поэтому выполните команду gpedit.msc, чтобы открыть редактор локальной групповой политики.

Перейдите в **Конфигурация компьютера, Административные шаблоны, Компоненты Windows, Службы удаленных рабочих столов, Узел сеансов удаленных рабочих столов, Лицензирование** (рис. 5.16).

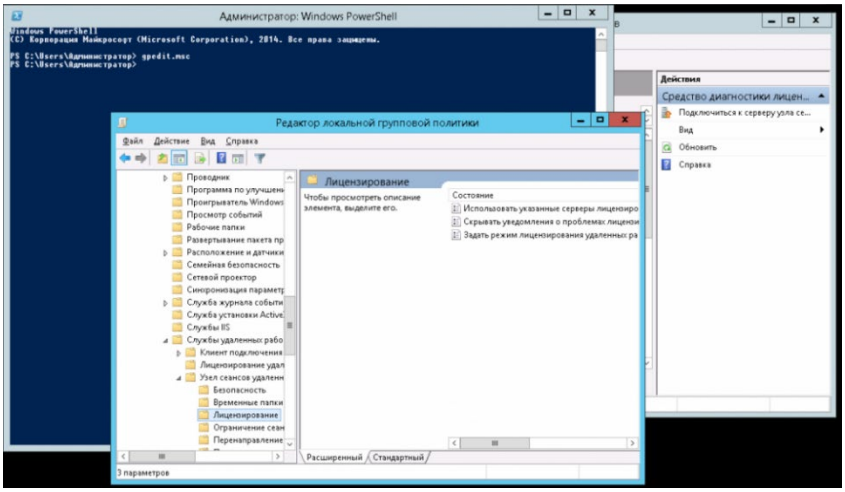

Рисунок 5.16. Редактор локальной групповой политики.

Далее необходимо открыть параметры «**Использовать указанные серверы лицензирования удаленных рабочих столов»**. В появившемся окне установить переключатель в положение «**Включено»** и указать, какой сервер лицензирования использовать. Мы будем использовать этот же сервер (рис. 5.17). Укажите имя сервера или его IP-адрес и нажмите кнопку **OK**.

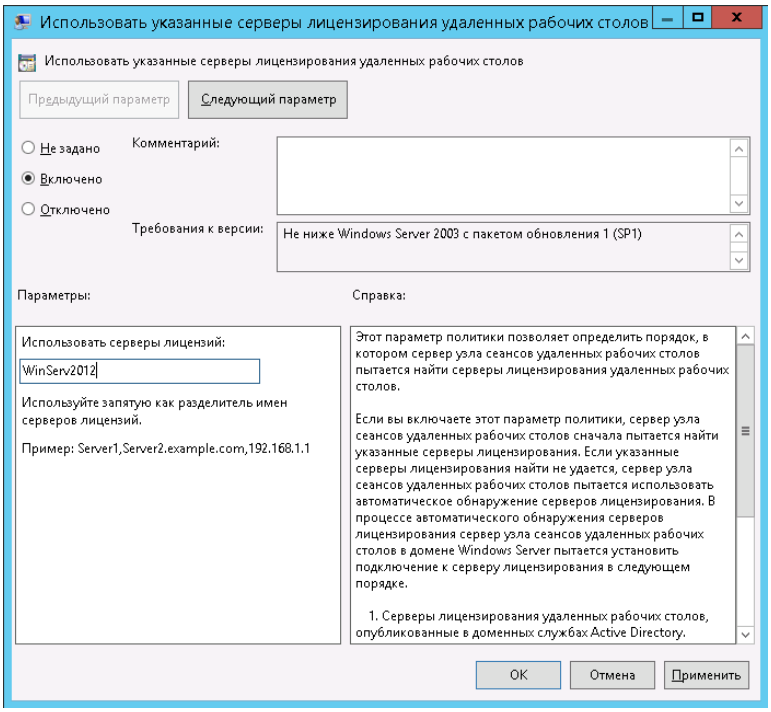

Рисунок 5.17. Параметры сервера лицензирования.

После проведенных операций необходимо перейти в параметры «**Задать режим лицензирования удаленных рабочих столов»** и перевести переключатель в положение **«Включено»,** также необходимо указать режим лицензирования сервера узла сеансов удаленных рабочих столов. Будет предоставлено два варианта – «На устройство» или «На пользователя». Предположим, имеется 10 лицензий. В случае выбора режима «На устройство» появится возможность создания неограниченного числа пользователей на сервере, которые смогут подключаться через удаленный рабочий стол только с 10 компьютеров, на которых установлены лицензии. В режиме «НА пользователя» на сервер смогут подключаться только 10 выбранных

пользователей, но с любых устройств. В большинстве случаев режим «На пользователя» более предпочтительный, поэтому рекомендуется выбирать его.

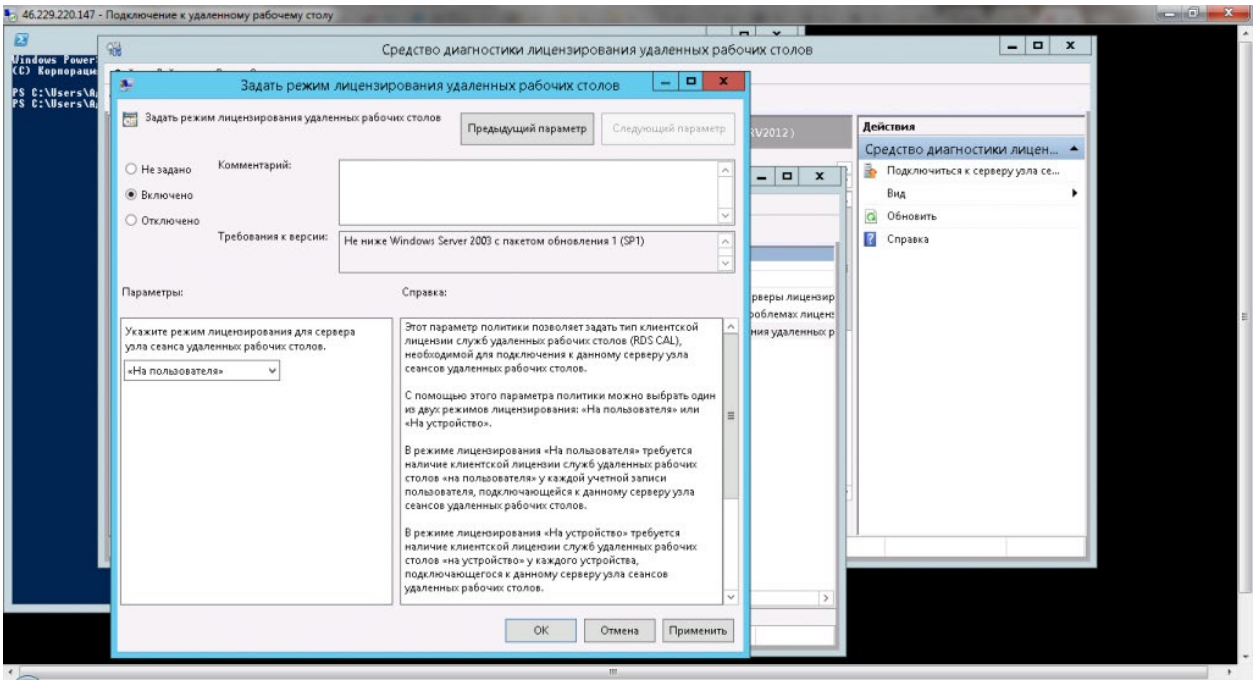

Рисунок 5.18. Выбор режима лицензирования.

В результате должны быть установлены такие параметры, как указано на рисунке 5.19. После чего необходимо закрыть окно редактора локальной групповой политики и перейти в окно средства диагностики лицензирования удаленных рабочих столов и нажать кнопку **обновить**. Средства диагностики лицензирования удаленных рабочих столов выдадут новую ошибку, которая сообщает о том, что сервер лицензирования не включен (рисунок 5.20).

Для запуска сервера лицензирования необходимо перейти в **«Диспетчер лицензирования удаленных рабочих столов»,** выбрать из списка нужный сервер, вызвать меню нажатием правой кнопки мыши и выбрать команду **«Активировать сервер»** (рисунок 5.21).

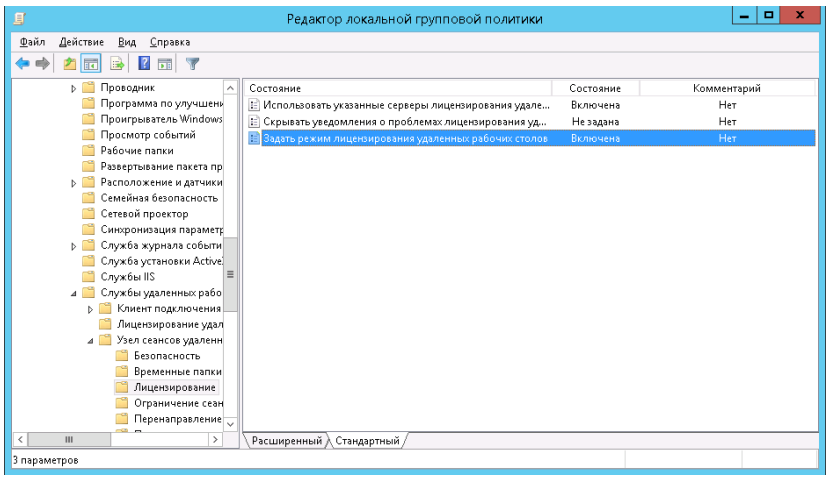

Рисунок 5.19. Результат установки параметров.

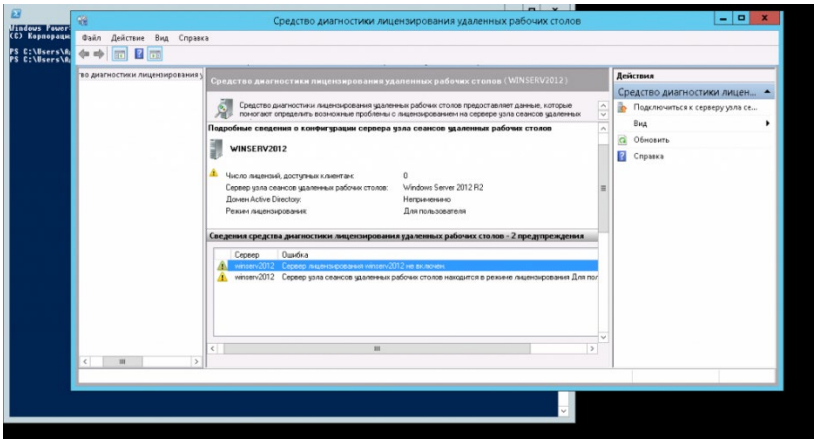

Рисунок 5.20. Новая ошибка.

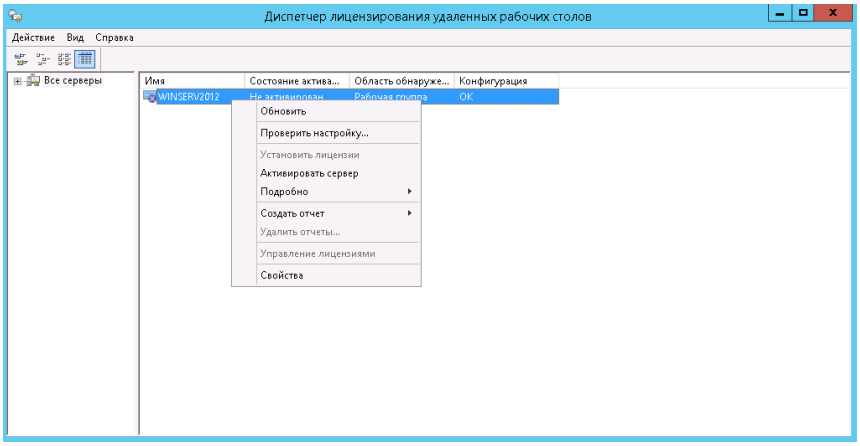

Рисунок 5.21. Диспетчер лицензирования удаленных рабочих столов.

После открытия окна **мастера активации сервера**, в котором нужно нажать кнопку **далее** — на первой странице мастера (рис. 5.22) и выбрать метод подключения. Рекомендуется оставить предложенный метод (рис. 5.23).

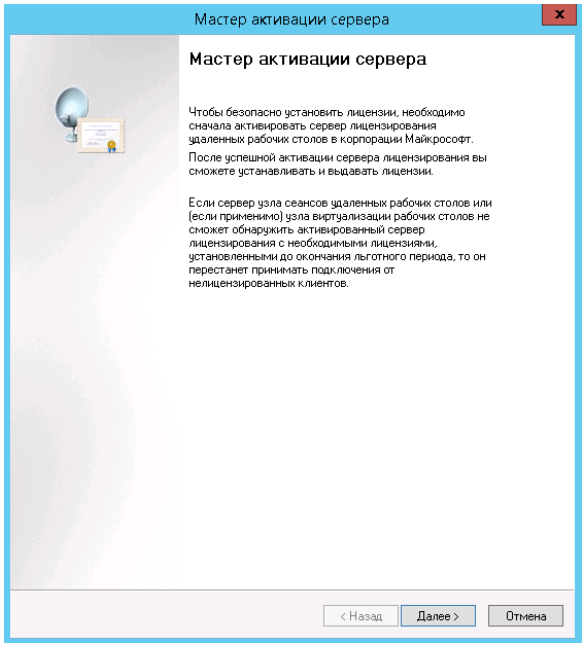

Рисунок 5.22. Мастер активации сервера.

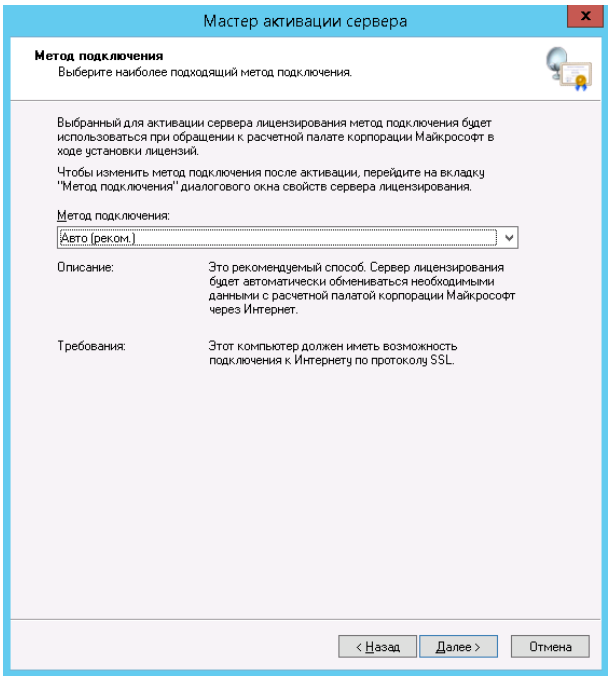

Рисунок 5.23. Выбор метода подключения.

Завершающим этапом является введение сведений об организации, после внесения которых необходимо нажать кнопку **далее.** После чего будет предложено внести дополнительные сведения об организации указывать которые не рекомендуется. Сервер лицензирования успешно активирован. Далее необходимо запустить мастер установки лицензий. Не выключая флажок **Запустить мастер установки лицензий** нажмите кнопку **Далее** как указано на рисунке 5.24.

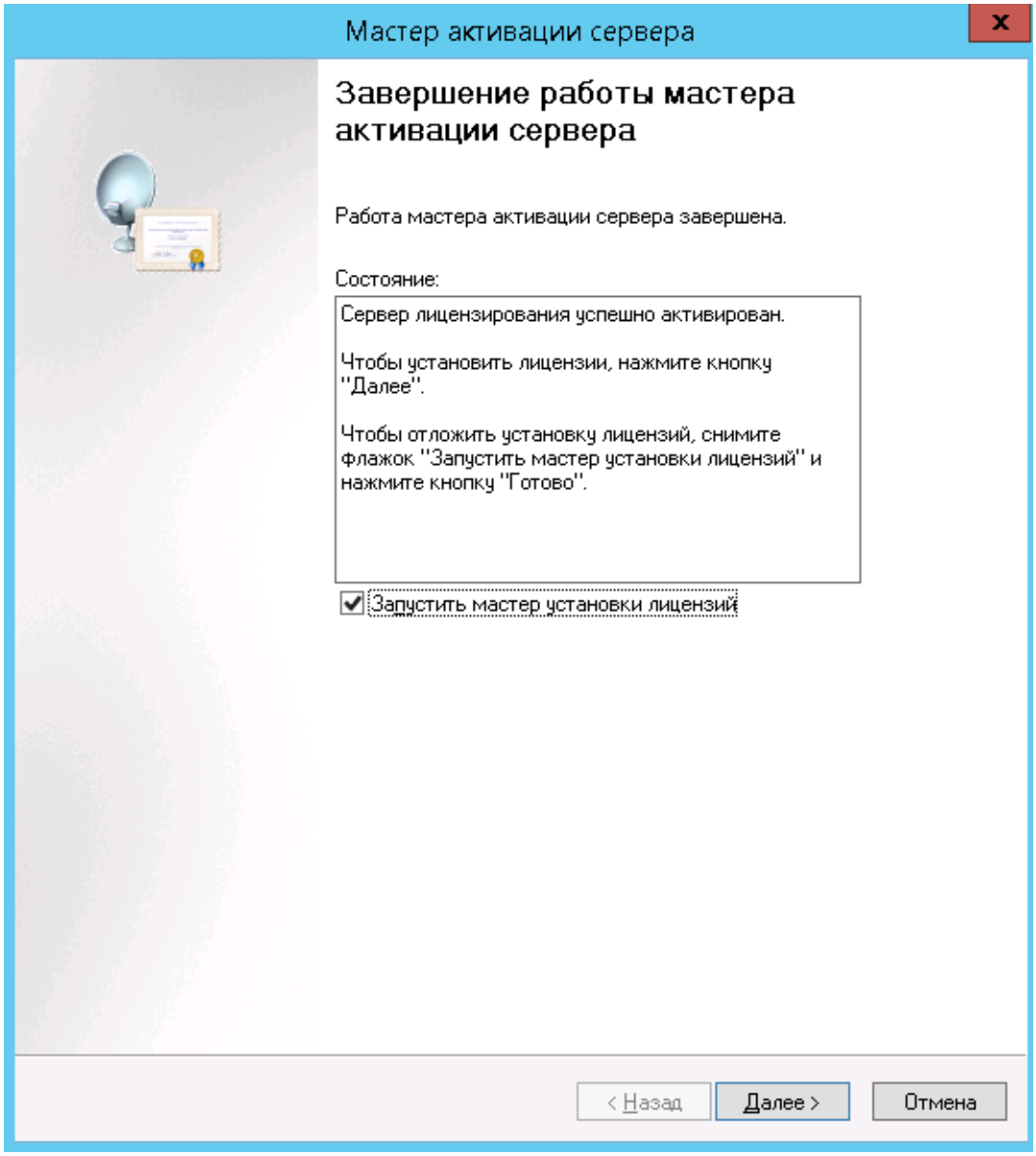

Рисунок 5.24. Сервер лицензирования активирован.

5.2.3. Установка лицензий службы удаленных рабочих столов

Результатом успешно проведенных мероприятий, описанных в предыдущем пункте, отобразится окно, сообщающее о запуске мастера установки лицензий (рис. 5.25).

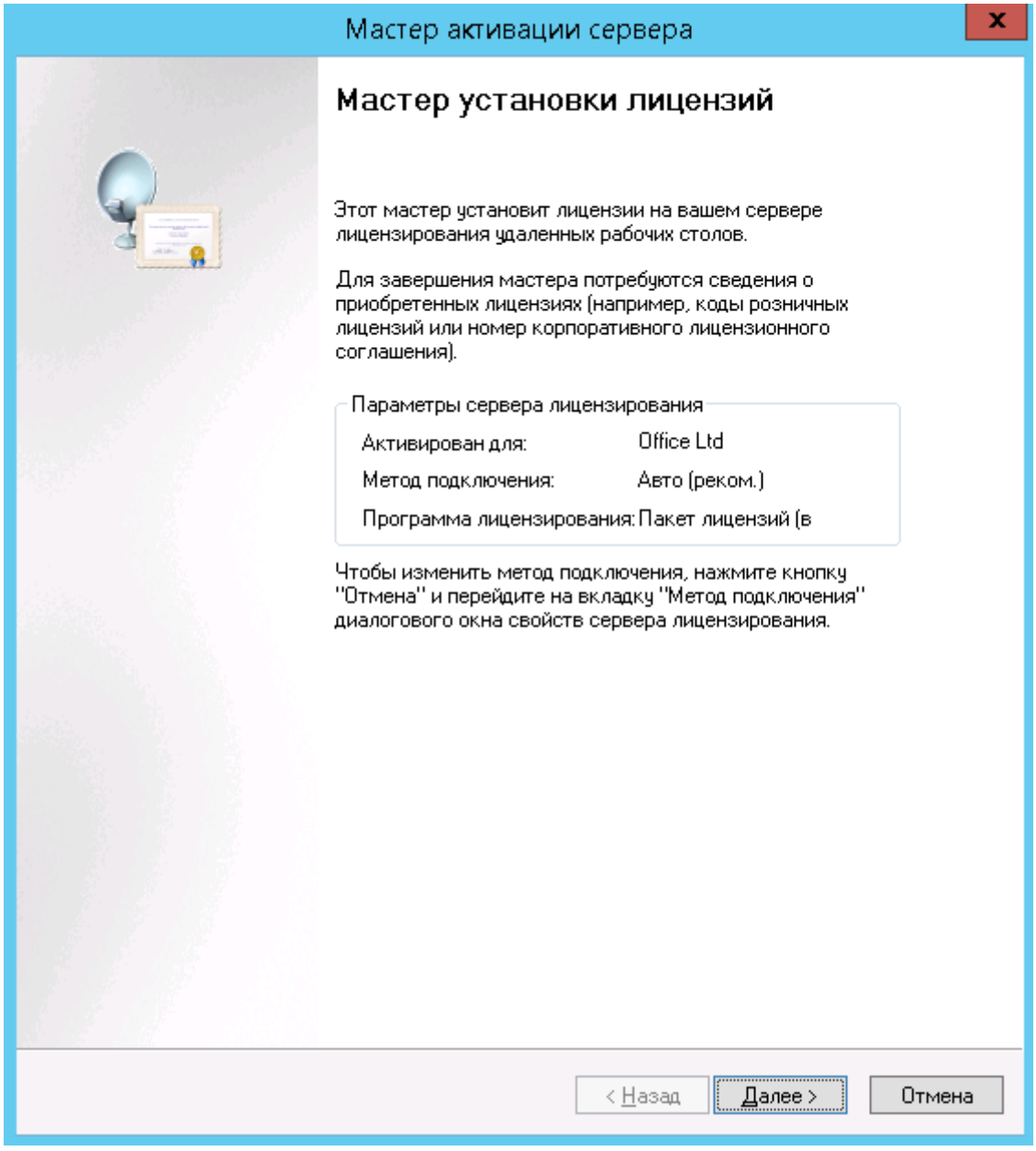

Рисунок 5.25. Мастер установки лицензий запущен.

Выберите нужную вам программу лицензирования. Здесь вы должны выбрать именно вашу программу лицензирования. В пособии из соображений

демонстрации настройки сервера мы будем использовать программу **Соглашение «Enterprise Agreement»** (рис. 5.26).

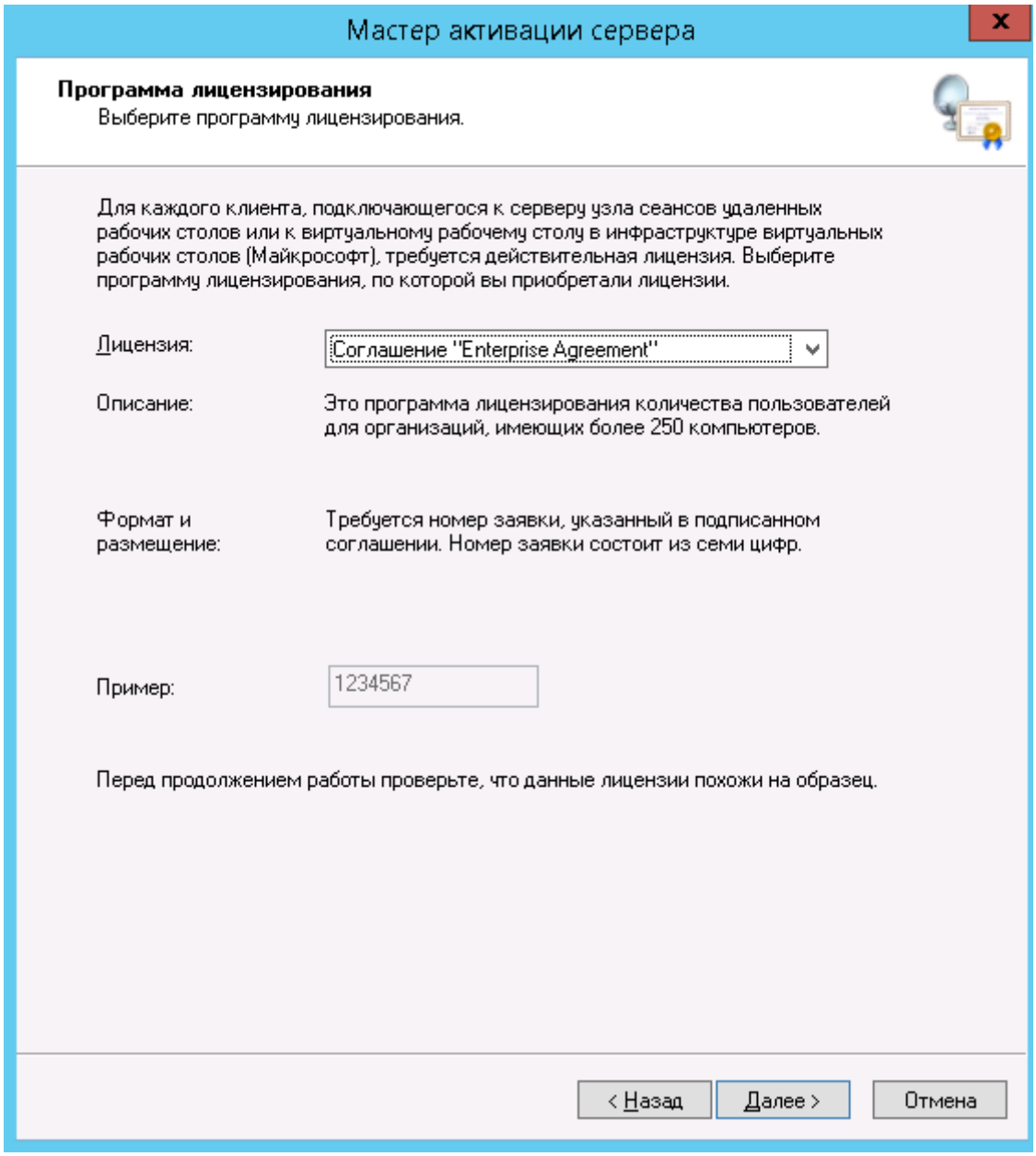

Рисунок 5.26. Выбор программы лицензирования.

Введите номер соглашения. Обычно он состоит из семи цифр. Нажмите кнопку **далее** (рис. 5.27).

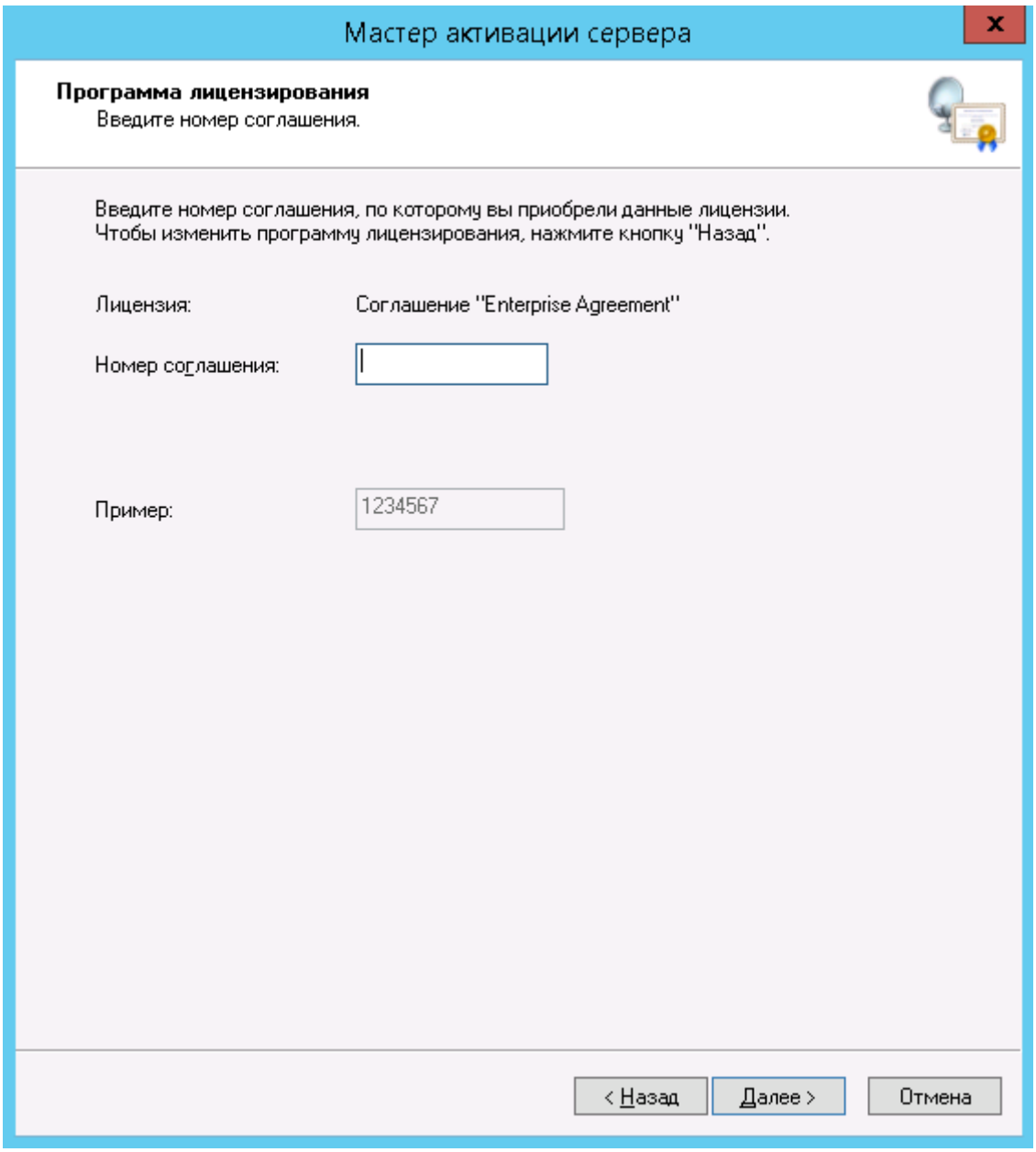

Рисунок 5.27. Ввод номера соглашения.

Далее нужно выбрать версию продукта (выбираем Windows Server 2012), тип лицензии и количество лицензий. Ранее мы выбрали тип лицензирования «На пользователя», поэтому здесь мы должны выбрать **Клиентская лицензия служб удаленных рабочих столов «на пользователя».** Введите число лицензий, доступных на данном сервере лицензирования.

Нажмите кнопку **далее** и дождитесь успешного завершения работы мастера установки лицензий. Если вы все сделали правильно, вы получите сообщение о том, что все запрошенные лицензии установлены.

Откройте **Диспетчер лицензирования удаленных рабочих столов**. Нужно убедиться, что сервер работает, а также просмотреть общее доступное число установленных лицензий. Вернитесь в **Средство диагностики лицензирования удаленных рабочих столов** и убедитесь, что нет никаких ошибок. Заодно вы увидите информацию об установленных лицензиях.

На этом настройка сервера терминалов завершена. Вам осталось только добавить пользователей, которым разрешен удаленный доступ к серверу, в группу **Пользователи удаленного рабочего стола** (Remote Desktop Users). Ведь по умолчанию RDP-доступ к рабочему столу сервера есть только у администраторов сервера.

Для подключения к терминальному серверу используется приложение **Подключение к удаленному рабочему столу**. Ничего сложного в его использовании нет — нужно указать IP-адрес сервера, имя пользователя и его пароль. При правильной настройке сервера будет выполнено подключение.

5.3. Настройка безопасности подключения к удаленному рабочему столу Windows Server 2016

5.3.1. Смена стандартного порта Remote Desktop Protocol

Начнем со стандартной меры — смены стандартного порта Windows 2016 RDP, что позволит предотвратить атаку всем известных портов (well-known ports).

Настройку порта можно осуществить с помощью реестра – HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TerminalServer\WinStations\RDP-Tcp\PortNumber.

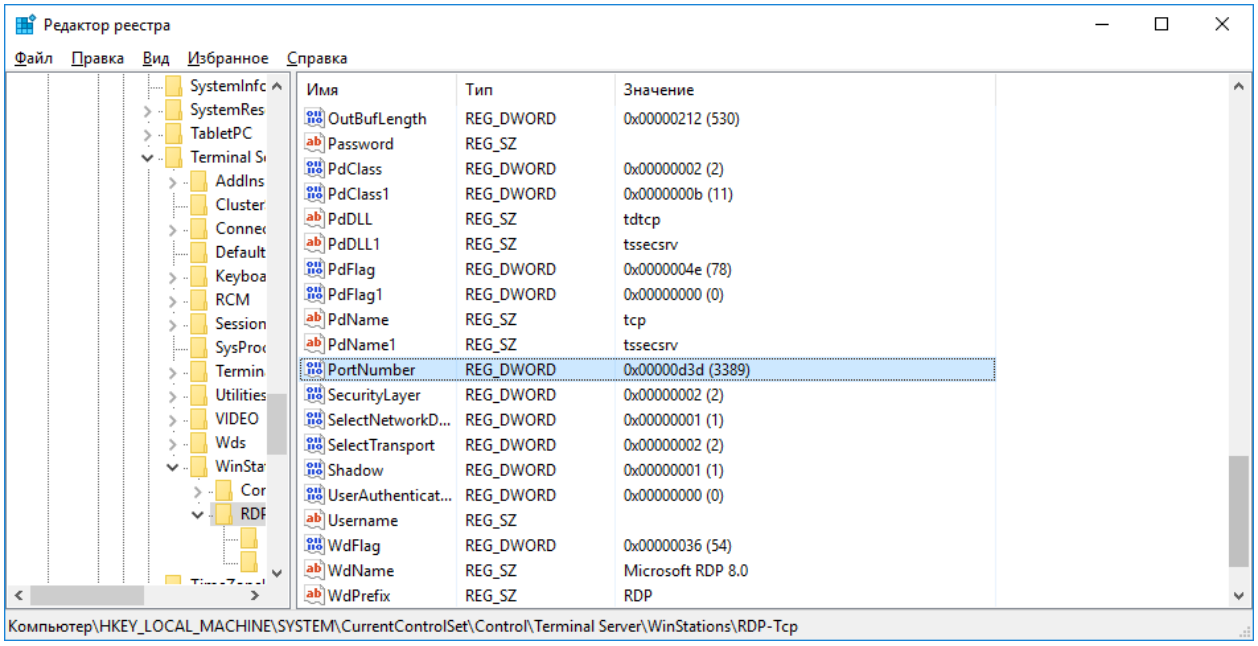

Рисунок 5.28. Смена порта RDP в Windows Server 2016 с помощью редактора реестра.

После этого запустите **Брандмауэр Windows** и на боковой панели слева выберите **Дополнительные параметры**. Далее — **Правила для входящих соединений** и нажмите кнопку **Создать правило** (на панели справа) как показано на рисунке 5.29.

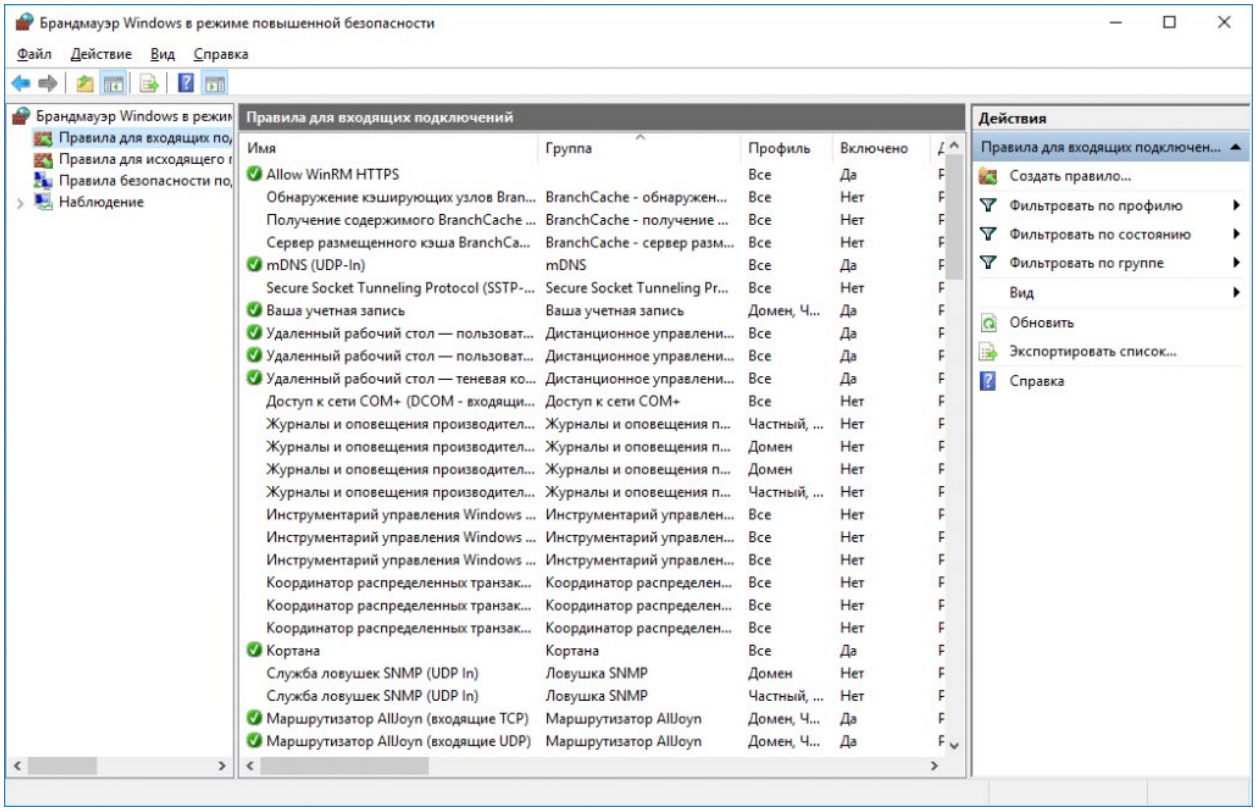

Рисунок 5.29. Правила для входящих подключений.

В появившемся окне выберите **для порта** и нажмите кнопку **Далее**. Затем выберите Определенные локальные порты и введите порт, указанный при редактировании реестра (рис. 5.30). В следующем окне мастера нужно выбрать **разрешить подключение** (рис. 5.31).

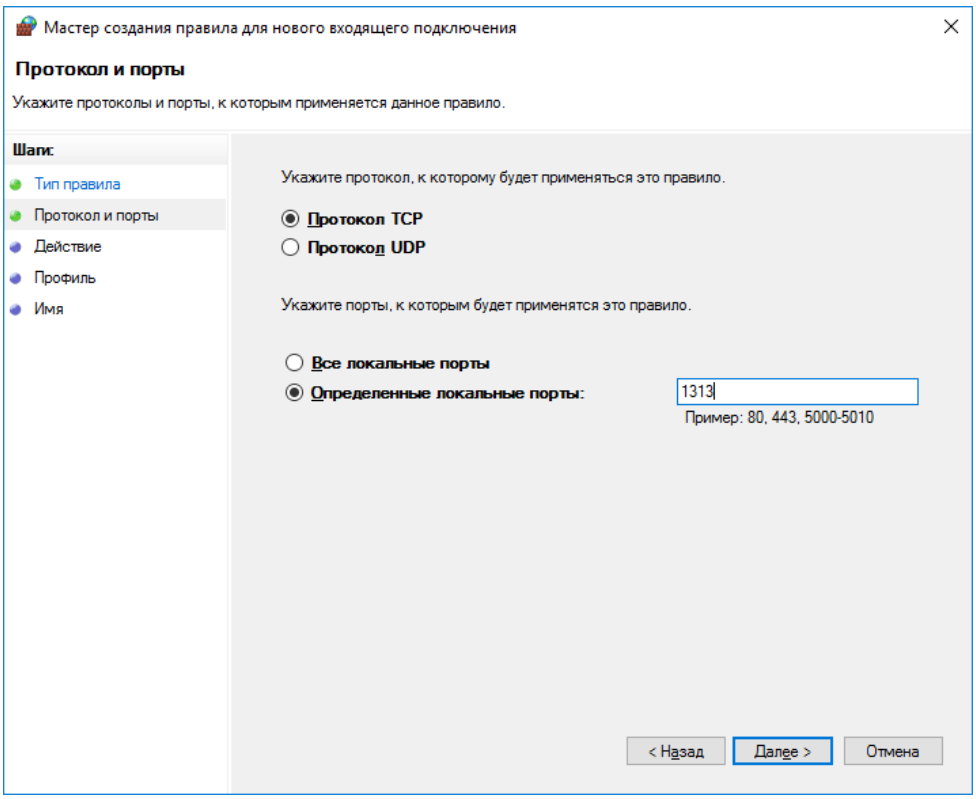

Рисунок 5.30. Задание нового порта.

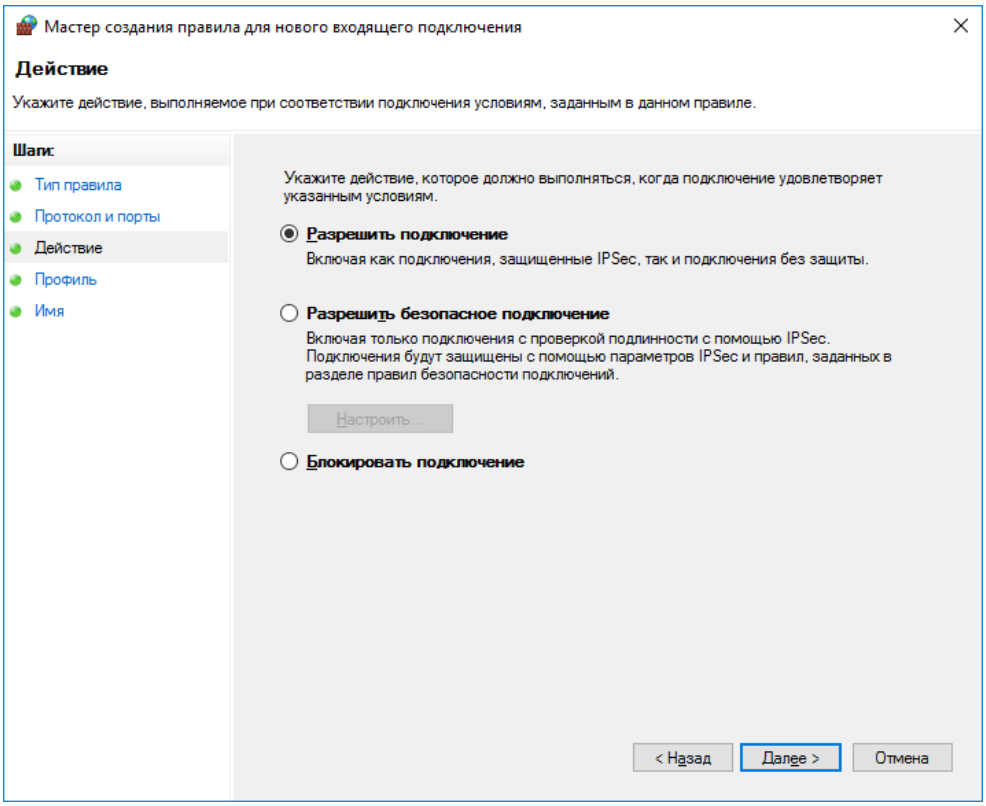

Рисунок 5.31. Разрешение подключения.

Собственно, на этом настройка безопасности RDP завершена. Отключитесь от сервера и установите новое соединение. Не забудьте в настройках RDP-клиента Windows Server 2016 указать новый порт: IP-адрес_сервера: порт.

5.3.2. Блокирование учетных записей с пустым паролем

Усилить RDP безопасность можно, запретив windows server подключаться учетным записям с пустым паролем. Для этого нужно включить политику безопасности «Учетные записи: разрешить использование пустых паролей только при консольном входе». Для этого необходимо выполнить следующие действия:

– открыть локальную политику безопасности (нажмите Win+R и введите команду secpol.msc);

– перейти в раздел **Локальные политики, Параметры безопасности**;

– дважды нажать на нужной нам политике и убедиться, что для нее задано значение **включен** (рис. 5.32).

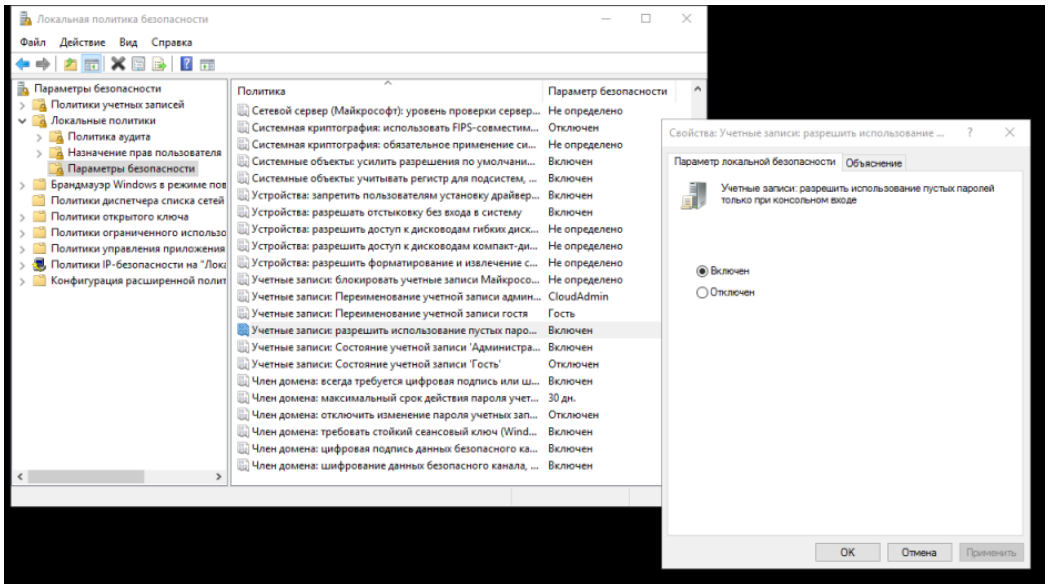

Рисунок 5.32. Включение политики безопасности в Windows 2016 RDP.

5.3.3. Настройка политики блокировки

Основная проблема в том, что по умолчанию Windows-server (даже 2016) не защищен от брутфорса, поэтому безопасность RDP в Windows 2016 пребывает не на очень высоком уровне. При наличии какого-либо сервиса, в частности, FTP, вас могут брутфорсить, пока не получат доступ к системе, перебирая огромное множество логинов и паролей. Именно поэтому необходима настройка временной блокировки пользователя после нескольких неудачных попыток.

Необходимый набор правил и настроек называется **Политика блокировки учетной записи** (Account Lockout Policy). Пока вы еще не закрыли окно **Локальная политика безопасности**, перейдите в раздел **Политики учетных записей**, **Политика блокировки учетной записи**. Установите **Пороговое значение блокировки** — 5 (максимум 5 неудачных попыток входа), Продолжительность блокировки учетной записи — 30 (на 30 минут учетная запись будет заблокирована после 5 неудачных попыток входа).

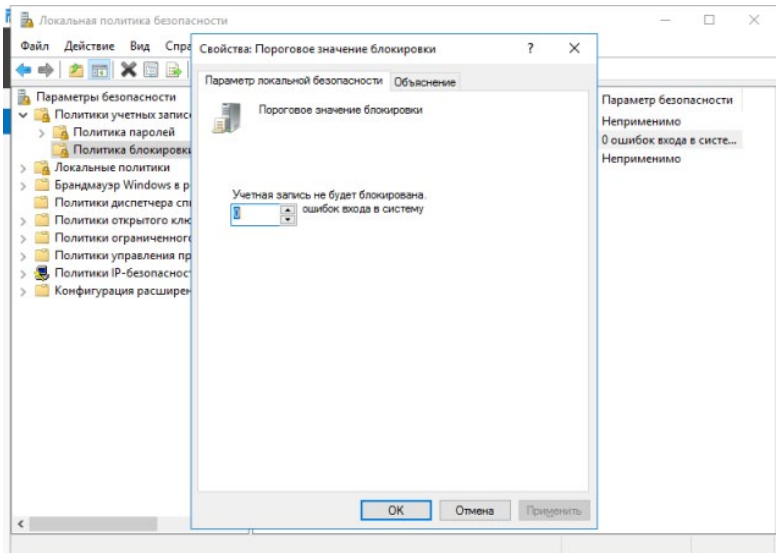

Рисунок 5.33. Настройка политики блокировки учетной записи.

5.3.4. Настройка службы шлюза служб терминалов

Служба «Шлюз TS (служб удаленных рабочих столов)» Windows Server 2016 разрешает сторонним пользователям реализовывать удаленное подключение к терминальным серверам и другим машинам частной сети с активированным протоколом удаленного рабочего стола. Служба обеспечивает RDP безопасность подключений за счет использования протокола HTTPS/SSL, что позволяет не заниматься настройкой VPN для шифрования трафика.

Служба позволяет контролировать доступ к машинам, устанавливая правила авторизации и требования к удаленным пользователям. Администратор сможет контролировать пользователей, которые могут подключаться к внутренним сетевым ресурсам, сетевые ресурсы, к которым могут подключаться пользователи, определять, должны ли клиентские компьютеры быть членами групп Active Directory и другие.

Порядок действий для установки службы «Шлюз TS»:

– открытие **Диспетчера серверов** и на главной его странице нажатие ссылки **добавить роли и компоненты;**

– нажатие **далее**, а затем выбор **установка ролей или компонентов**;

– выбор нужного сервера из пула серверов;

– выбор **службы удаленных рабочих столов** и нажатие **далее** несколько раз, до достижения страницы **выбор служб ролей;**

– выбор службы ролей **шлюз удаленных рабочих столов** (рис. 5.34);

– нажатие кнопки **добавить компоненты;**

– нажатие **далее** для завершения процедуры установки. Когда все будет готово для установки выбранных компонентов, необходимо выбрать **установить** (рис. 5.35).

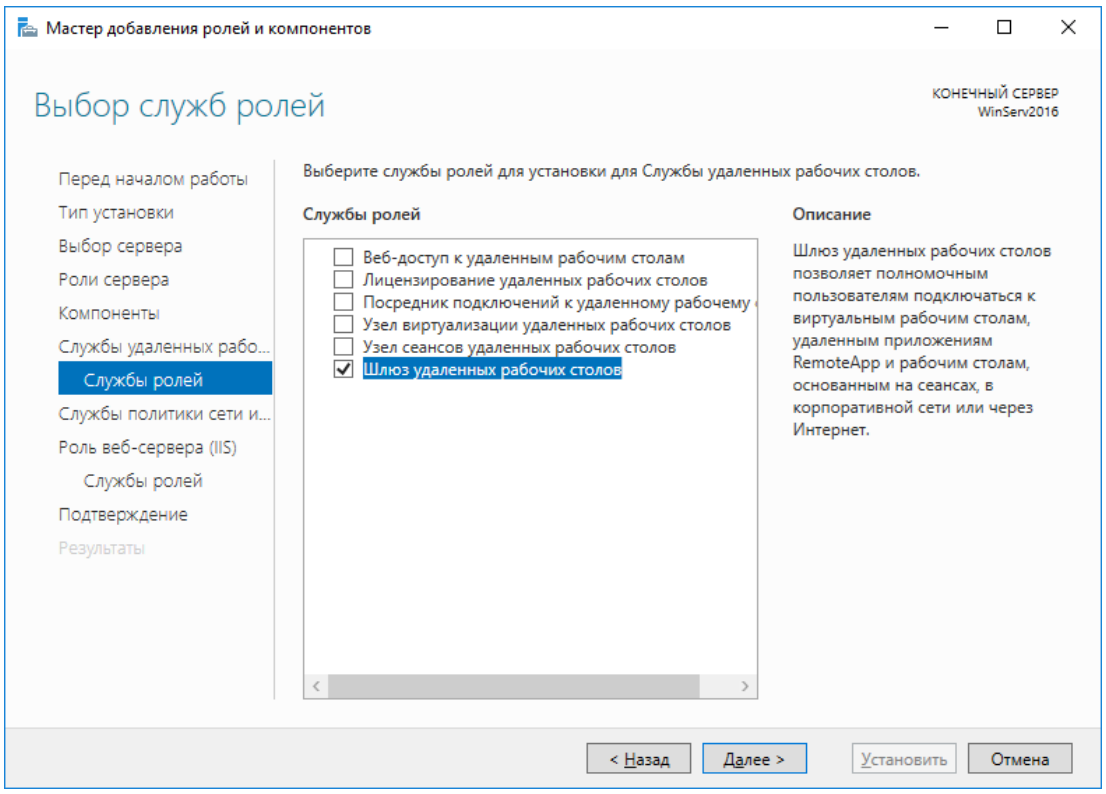

Рисунок 5.34. Установка службы ролей.

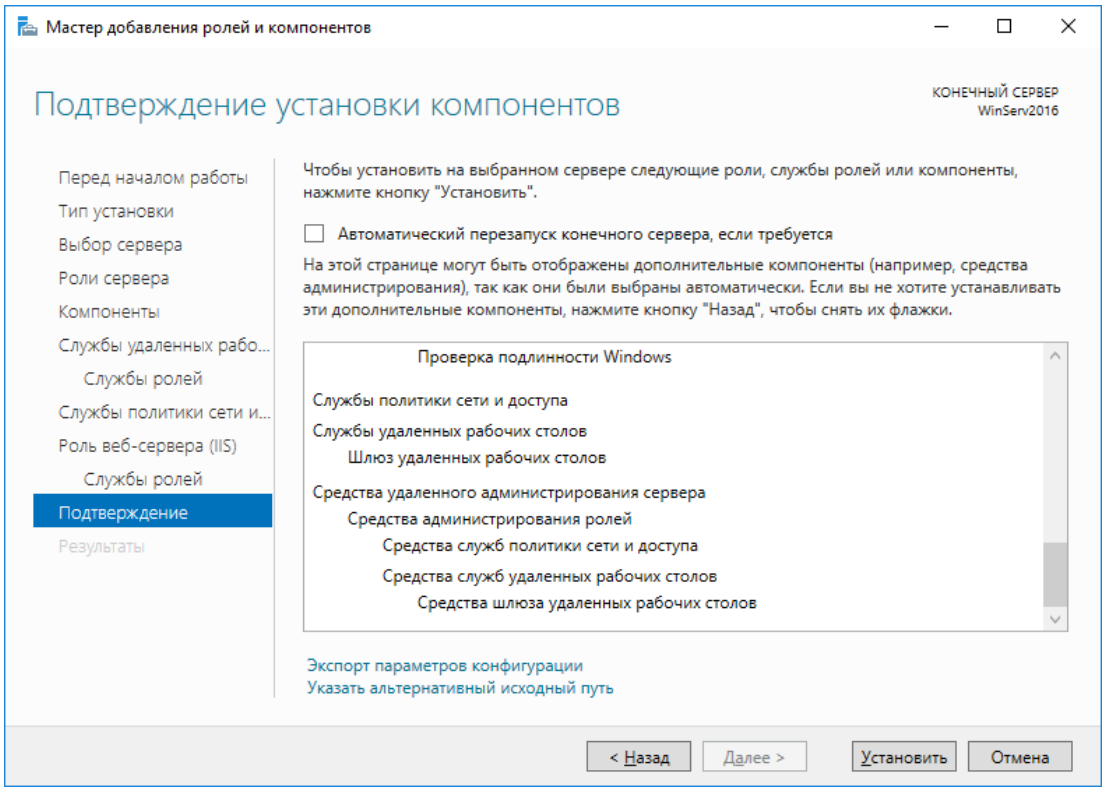

Рисунок 5.35. Подтверждение установки компонентов.

Далее нужно создать сертификат для шлюза служб удаленных рабочих столов. Запустите **диспетчер служб IIS**, выберите пункт **сертификаты сервера**. На панели справа выберите **создать сертификат домена**. Введите необходимые сведения (рис. 5.38).

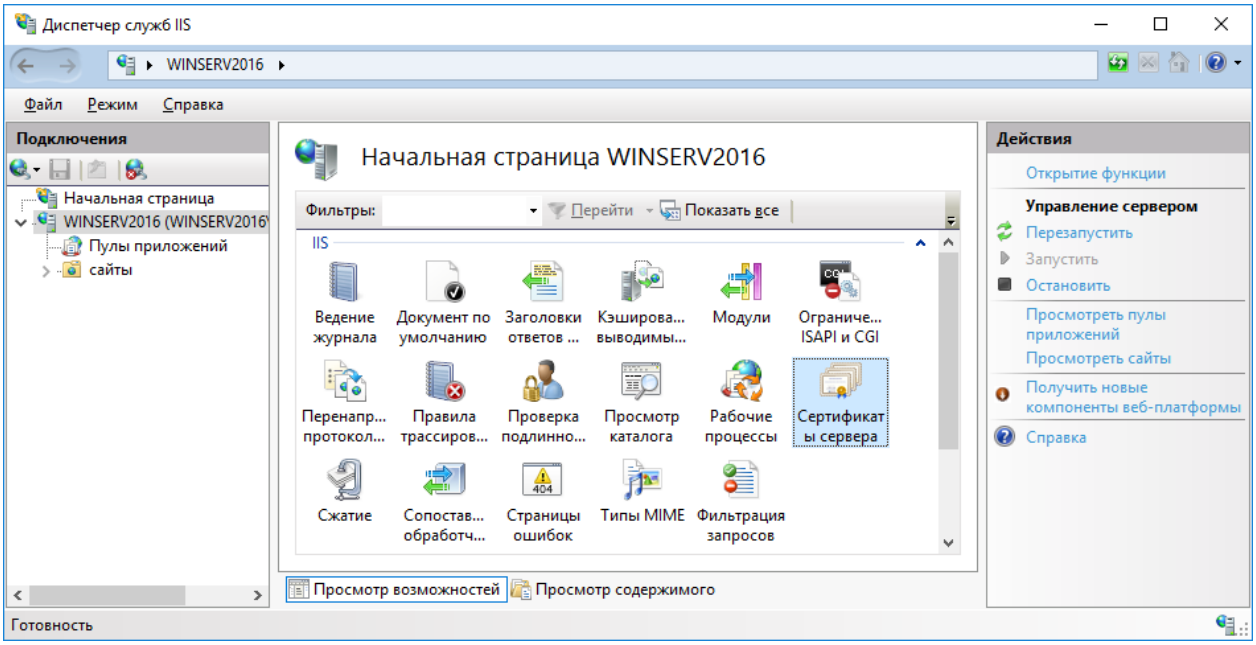

Рисунок 5.36. Диспетчер служб IIS.

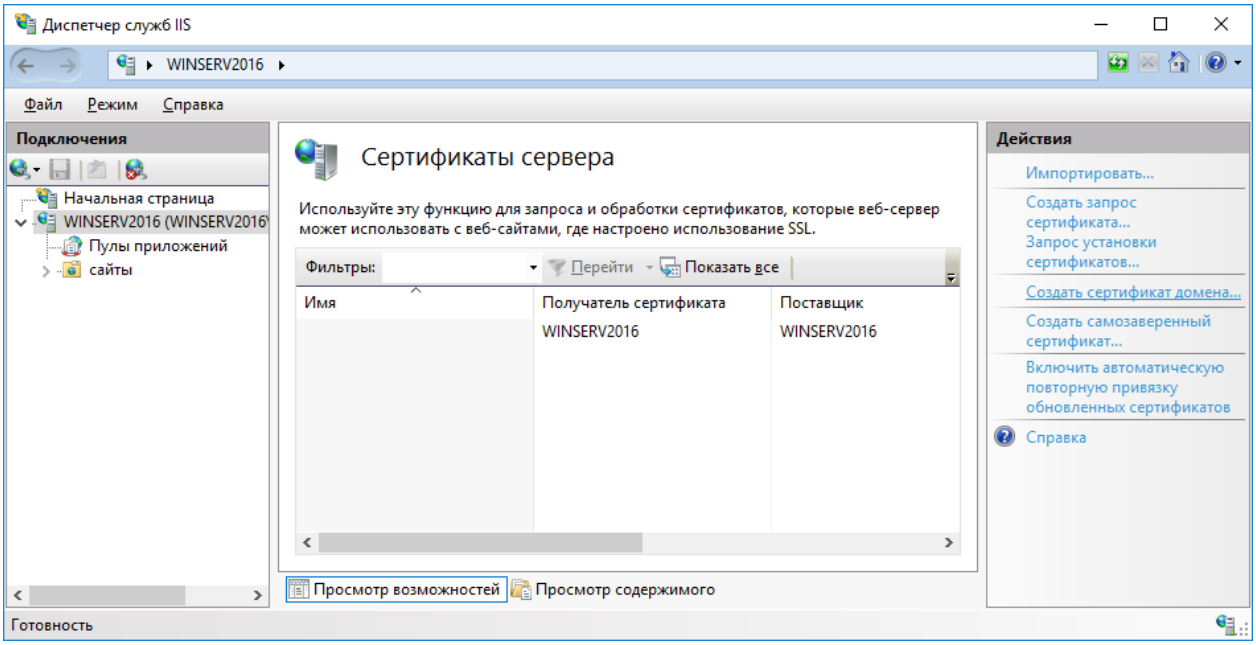

Рисунок 5.37. Сертификаты сервера.

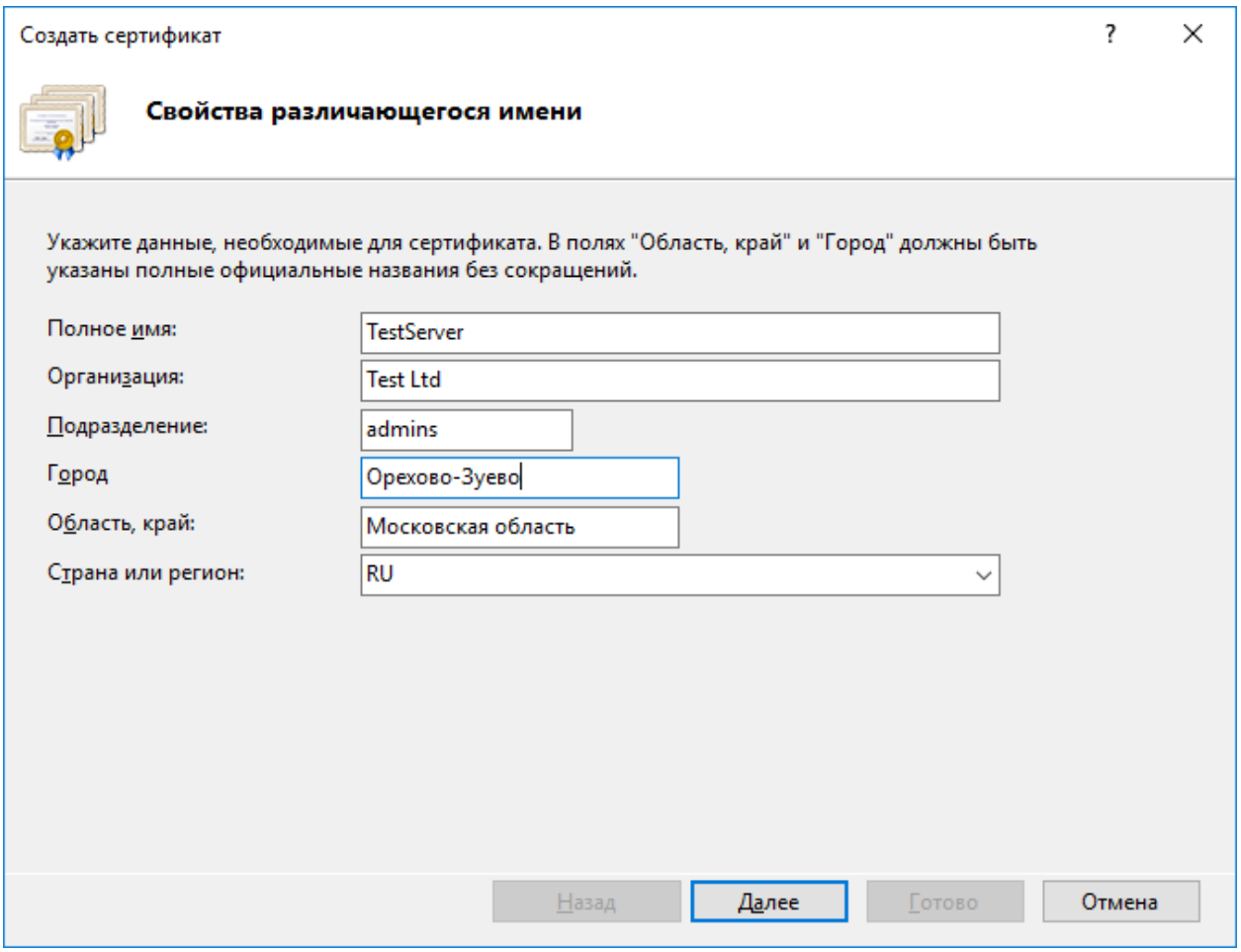

Рисунок 5.38. Информация о сертификате.

Далее нужно выбрать центр сертификации, который подпишет сертификат, и нажать кнопку **готово**. Сертификат появится в списке сертификатов.

Следующий шаг — настройка шлюза TS на использование сертификата. Вернитесь к **диспетчеру серверов** и, используя меню **средства**, запустите **диспетчер шлюза удаленных рабочих столов**. Обратите внимание, что есть одна ошибка — **сертификат сервера еще не установлен или не выбран**. Собственно, все, что остается сделать — нажать ссылку рядом **просмотр и изменение свойств сертификата** и выбрать ранее созданный сертификат, нажав кнопку **импорт сертификата** (рис. 5.40). В этом окне также можно создать самозаверяющийся сертификат.

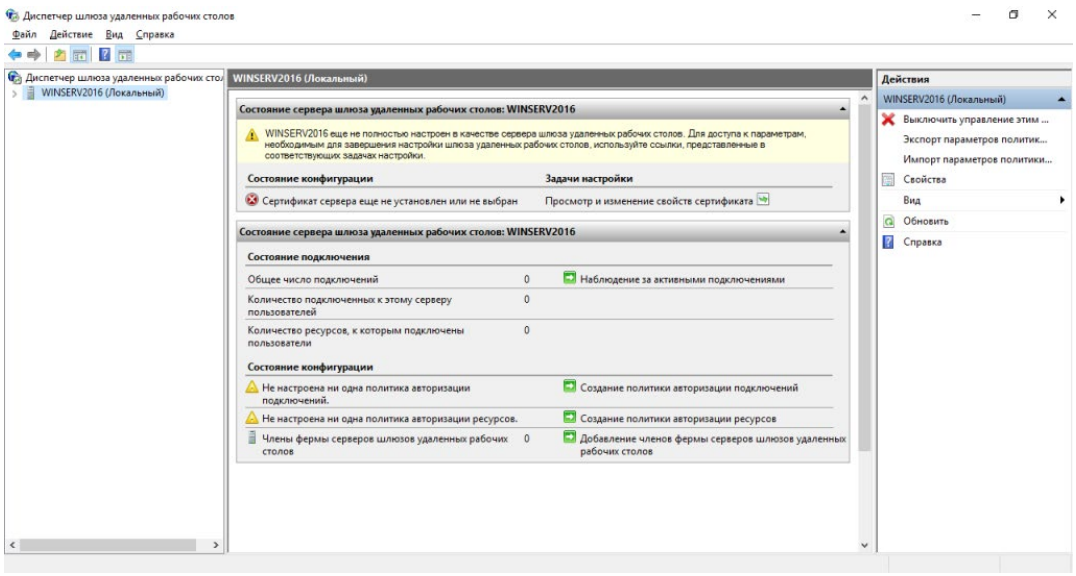

Рисунок 5.39. Диспетчер шлюза удаленных рабочих столов.

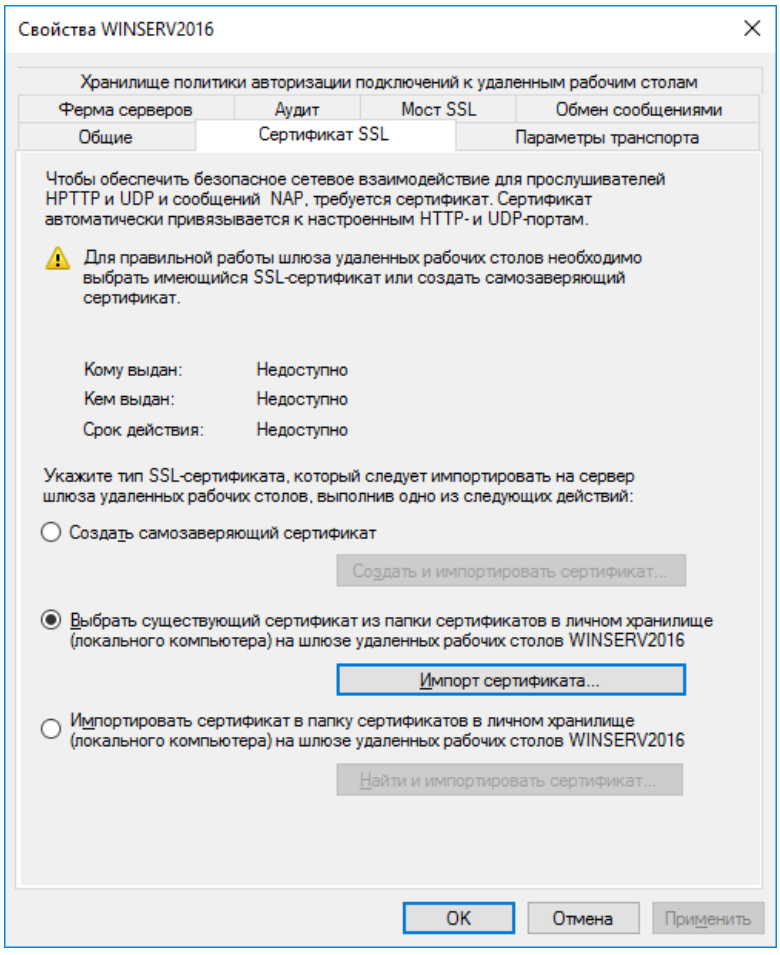

Рисунок 5.40. Выбор сертификата SSL.

5.3.5. Использование протокола SSL/TLS для защиты RDP

Если соединение с RDP-сервером реализовывается не через VPN, для обеспечения защиты подключений рекомендуется активировать SSL/TLS-туннелирование соединения.

Опцию RDP через TLS можно активировать через набор правил и настроек защиты сервера удаленных рабочих столов. Введите команду gpedit.msc и перейдите в раздел **конфигурация компьютера, административные шаблоны, компоненты Windows, службы удаленных рабочих столов, узел сеансов удаленных рабочих столов, безопасность**. Откройте политику **требовать использования специального уровня безопасности для удаленных подключений по протоколу RDP**. Выберите значение **включено** и выберите уровень безопасности **SSL** (рис. 5.41).

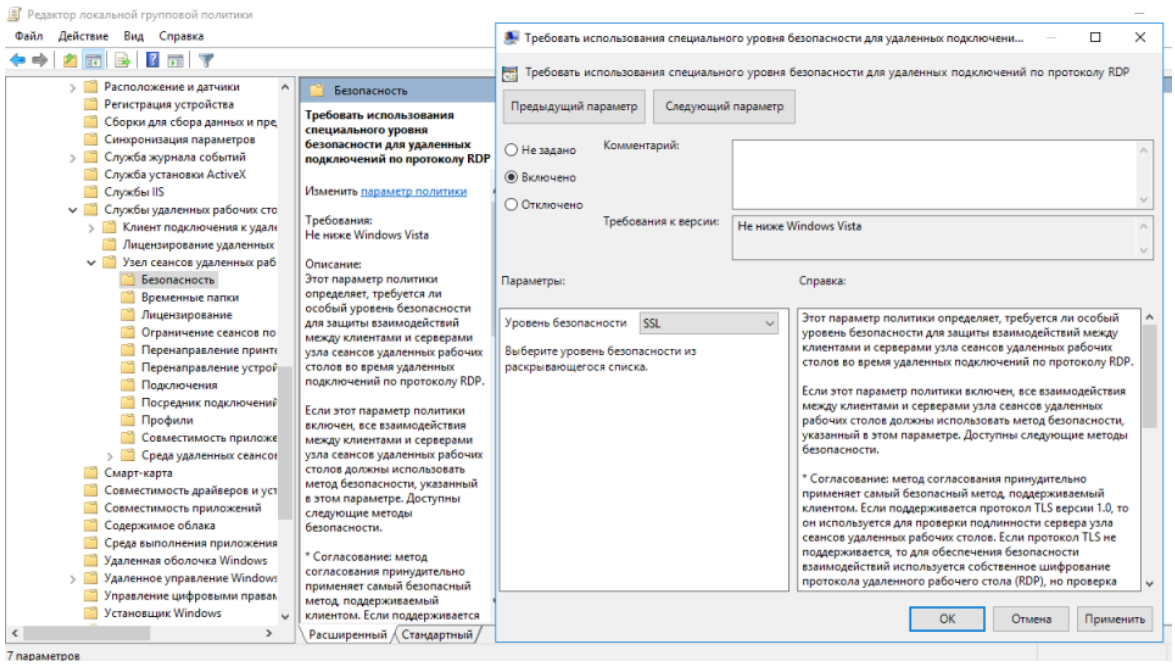

Рисунок 5.41. Включение RDP через TLS.

5.3.6. Отличия выделенных серверов от виртуальных

VDS/VPS – вид хостинга, при котором на одной физической машине находится множество выделенных хостов. То есть, провайдер предоставляет клиенту в пользование свои ресурсы. Virtual Dedicated Server можно назвать оптимальным решением для проектов, для которых выделенный сервер является дорогим приобретением.

На одном хосте располагается множество проектов, для каждого из них отведено рабочее пространство. При этом ресурсы (то есть процессор и оперативная память) у всех сайтов общие. Такое решение достаточно безопасное, так как провайдер может изолировать данные разных клиентов друг от друга.

Среди главных достоинств VPS отмечают:

- разнообразные возможности настройки;
- высокий уровень безопасности;
- возможность быстрого изменения оперативной памяти;
- выбор тарифного плана в зависимости от потребностей проекта.

Обычно сайты, создаваемые с нуля, первое время работают именно на VDS. Такое решение вполне логично, так как позволяет экономить и при этом обеспечивает стабильную работу проекта. Однако с постепенным ростом контента и баз данных имеющихся мощностей оказывается недостаточно, поэтому многие переходят на индивидуально выделенный хост.

Dedicated server, или DS – это отдельная физическая машина, которую клиент может арендовать у провайдера. То есть, все ресурсы идут на один проект, пользователь получает доступ к удаленной настройке BIOS.

Решение обычно используется для крупных сайтов, а также проектов, требующих индивидуальную настройку серверов. Среди главных достоинств:

- непосредственный доступ ко всем мощностям хостинга.
- возможность самостоятельно корректировать конфигурацию и ПО оборудования.

– отслеживание всех действий пользователей, процессов и файлов.

– возможность самостоятельной загрузки ПО и установки только важных программ.

Dedicated подходит для проектов, которым требуется большое количество ресурсов. Например, если база данных потребляет много вычислительных ресурсов. Такой вариант хостинга наиболее стабилен в работе, так как он единовременно используется только для одного сайта. Также клиент может выполнить индивидуальные настройки оборудования, установить ПО и приложения и т. д. (но чаще всего это происходит через провайдера).

Давайте подробнее рассмотрим отличия выделенных серверов от виртуальных. Хоть эти два названия и похожи, но они имеют множество различий и применяются для совершенно противоположных проектов.

Контроль над управлением. Это – самая главная разница между данными хостингами. При использовании VDS клиент может самостоятельно получить доступ к управлению сервером, установить требуемое ПО и приложения, контролировать список пользователей и т.д. DS же предполагает, что контроль над установкой программ и настройками системы находится у провайдера. Но при этом нет ограничений на используемые приложения и программы.

Предоставляемые ресурсы. Использование виртуального сервера гарантирует постоянную доступность ресурсов. Клиент всегда имеет определенный объем дискового пространства, и в большинстве случаев данные других пользователей никак не влияют на загрузку. Объем мощностей напрямую зависит от тарифа, выбранного клиентов. При желании всегда можно увеличить количество выделенных ресурсов, выбрав новый тарифный план. На выделенном хостинге размер ресурсов ограничен возможностями виртуальной машины. Пользователь получает доступ ко всем мощностям и может применять их для любых данных.

Ограничения. Выделенный хост позволяет разместить практически бесчисленное количество сайтов, баз данных и файлов, тогда как при использовании VPS клиент имеет ограниченные ресурсы. Конечно, есть

возможность перейти на другой тариф и увеличить мощности, это повысит цену обслуживания. Кроме того, рано или поздно расширение ресурсов сделать не получится, так как на каждого клиента провайдер может выделить только определенную часть мощностей.

Рассмотрим отличия выделенных серверов от виртуальных на конкретных примерах и расскажем, как выбрать решение в зависимости от ситуации.

«*Требуется хранить много данных*». Для крупных проектов, в которых используются большие базы данных и объемы информации, потребуется много дискового пространства. Физическая машина в этом случае выигрывает у виртуальных технологий. Кроме этого, вы сможете собрать собственный сервер с необходимой конфигурацией. Это гарантирует сохранность информации и работоспособность приложений в любое время.

«Предполагается постоянная загрузка процессора». Некоторые процессы (например, кодирование видео) могут загрузить процессов на 100%. При использовании VDS могут возникнуть проблемы, так как это наверняка будет мешать вашим «соседям». Следовательно, потребуется Dedicated. Однако если решение выходит за рамки, то можно поискать подходящий виртуальный хостинг. Для начала оцените предполагаемое время загрузки процессора – наверняка, «тяжелые» задачи не будут совершаться круглые сутки, следовательно, вы не создадите проблем для других пользователей.

«*Требуется много IP-адресов*». Виртуальные серверы позволяют подключить в среднем до 4 IP, однако можно заказать дополнительное подключение блоков. Аналогичная ситуация и с Dedicated, поэтому при выборе фактор не имеет важности, нужно делать упор на других технических характеристиках.

«*Потребуется много RAM*». В этом случае все зависит от того, как много оперативной памяти нужно. Если у вас небольшой или средний проект, то вам хватит мощностей виртуального сервера. В остальных случаях рекомендуется использовать физическое оборудование.

«*Необходимы частые бэкапы*». Резервное копирование данных лучше производить на отдельный хост – и для виртуального, и для выделенного хостинга это является дополнительной платной услугой.

«*Есть необходимость в высокочастотном процессоре*». Для ряда программ и приложений могут потребоваться процессоры с высокой тактовой частотой. На VPS используются процессоры со средними техническими показателями, поэтому лучше выбирать Dedicated.

«*Сервер потребовался срочно*». В этом случае выбор зависит от того, насколько «срочным» должно быть решение. Если вы уже используете сервер у другого провайдера, и он внезапно упал, то требуется поднять новый в минимальные сроки. В этом случае лучше выбирать виртуальный хост с виртуализацией KVM, так как вы получите готовую ИТ-инфраструктуру за несколько минут. Если вы можете подождать несколько часов, то подойдет Dedicated.

«*Требуется встроенная DDoS-защита*». Обе технологии предполагают наличие встроенной защиты, поэтому стоит руководствоваться другими критериями выбора.

«*Мощности будут использоваться в рабочие часы*». В этом случае вам должно хватить VDS. Плюс такой технологии в том, что есть возможность воспользоваться почасовыми тарифами и платить только за реально потраченные ресурсы. Использование выделенного хоста будет обходиться в разы дороже, что может быть критичным для небольших проектов.

«*Подключается специфическое оборудование*». В таких случаях рекомендуется использовать только физические выделенные серверы. Такая технология позволит вам защитить приложения USB-ключом или использовать арендованные порты на коммутаторе. Обеспечить высокую безопасность и скорость работы в это случае сможет только DS. Однако если вам требуются графические ускорители, то их можно использовать и на виртуальном хосте.

«*Требуются разные серверы под мелкие задачи*». Очень часто необходимо создание множество небольших серверов для решения мелких задач (например,

при парсинге данных с сайтов). Используя облачную технологию, можно запустить сразу несколько виртуальных машин, а после – удалить или приостановить ненужные серверы. При применении Dedicated это сделать проблематично.

У каждого сервера есть свои достоинства и недостатки, поэтому нельзя говорить о том, что какое-то решение подходит для сайтов больше. Выбор стоит делать, ориентируясь на цели и задачи вашего проекта, предполагаемых объемов информации, необходимого программного обеспечения и функционала. Если ваш сайт пока не предполагает использования больших баз данных и вам не нужно много ресурсов, то возможно выбрать виртуальный выделенный сервер. И, наоборот, для проектов с большей нагрузкой на хостинг и объемами ресурсов потребуется физическое оборудование.

5.4. Информационная безопасность ЦОД

Безопасность дата-центра – один из основополагающих факторов, которым руководствуются клиенты при поиске провайдера. Этот факт ничуть не удивляет, так как защита баз данных и файлов является критически важной для любого проекта.

Давайте разберемся, как обеспечивается информационная безопасность (ИБ) центров обработки данных, какие объекты защищаются в первую очередь и какие технологии для этого используются.

Обеспечение информационной безопасности ЦОД представляет собой комплекс различных технических мер, которые помогают предотвратить утечку данных и оградить файлы от несанкционированного доступа. И самыми «защищаемыми» компонентами в этом случае являются:

– ресурсы (то есть, непосредственно сами данные);

– действия по сбору, переработке, хранению и последующей передаче информации;

– пользователи сервера;

– ИТ-инфраструктура, включая все технические и программные инструменты обработки.

Составление политики безопасности информации требуется построения особой модели угроз и нарушителей. То есть необходимо составить список факторов, которые несут угрозу для ИТ-инфраструктуры и нормальной работы ЦОД.

Угрозой для дата-центра могут стать самые различные явления:

– природные бедствия;

– нарушения сроков поставок оборудования;

– террористические акты;

– события техногенного или социального характера;

– сбои и отказы программных и технических средств;

– внутренние нарушения ИБ сотрудниками и т. д.

Этот список можно продолжать бесконечно, так как безопасность ЦОД нарушается даже при банальных перебоях в электропитании.

Анализ рисков включает в себя не просто выявление возможных угроз, но и оценку масштабов их последствий. Это поможет составить план действий для обеспечения ИБ, распределить задачи по важности и спланировать бюджет для покупки необходимых технических и программных средств.

ИБ организуется с учетом требований законодательных стандартов, нормативных документов и прочих регуляторов. Стоит отметить, что обеспечение безопасности не является дополнительным источником дохода, а лишь минимизирует риски и позволяет повысить уровень обслуживания клиентов.

Большой риск для работы ЦОДа несут перебои в электропитании и возможные отказы системы охлаждения. К счастью, подобные угрозы удалось полностью нейтрализовать.

Происходит это за счет внедрения «умной» системы управления, которая объединяет в себе новейшие системы охлаждения, вентиляции и

электропитания, совмещенные с контроллерами для круглосуточного мониторинга.

Если в дата-центре реализуется услуга аренды, то помимо информационной безопасности, потребуется физическая защита оборудования. Важно предотвратить несанкционированный доступ к оборудованию и оградить его от внешнего воздействия.

Для этих целей применяются огороженные части помещения, которые обязательно находятся под постоянным видеонаблюдением. Также организовывается система пожаротушения, которая самостоятельно активируется в случае начинающегося пожара и задымления.

Сейчас стали внедрятся и более совершенные технологии защиты. Например, сервис контроля и управления стал полностью интеллектуальным, а сам доступ обеспечивается при сканировании биометрии.

Информационная безопасность ЦОДов, обеспечиваемая различными датчиками слежения и умным видеонаблюдением, к сожалению, не позволяет решить абсолютно всех проблем. Главный фактор риска остается – это специалисты, допущенные к работе в дата-центре.

На функционирование центра обработки данных может повлиять нецелевая эксплуатация системы персоналом. Для исключения подобных факторов используются специальные системы управления.

Дополнительной защиты требует и сам персонал, так как именно работники ЦОДа являются уязвимым местом во всей системе безопасности. К сожалению, нередки случаи, когда преступники начинают использовать способы социальной инженерии. Минимизация рисков в этом случае происходит за счет обучения персонала навыкам ИБ.

Дополнительную защиту от внешних угроз (например, от взрывов и пожаров в здании) обеспечивают комнаты и сейфы безопасности. В них серверное оборудование надежно защищено от любых угрожающих факторов.

Большинство дата-центров сегодня предоставляют услуги не только «колокейшн», но и аренды облачных ресурсов. Следовательно, требуется обеспечивать защиту нескольких каналов связи.

Атаки на cloud-технологии сегодня происходят все чаще – почти 50% ЦОДов сталкиваются с ними каждый год. Опасность DDoS-атак заключается в возможной установке вирусов-шифровальщиков и блокировке инфраструктуры компании.

Наиболее частыми являются угрозы взлома, кражи или удаления данных, которые хранятся на облачных сервисах.

Защита информации достигается при помощи инструментов для выявления и устранения вредоносного кода. Также многие ЦОД используют системы с функциями IPS.

Многие российские data-центры применяют внешние приложения, позволяющие перераспределять трафик на другие узлы, а затем производить его фильтрацию в облаке.

Для поддержания безопасности ЦОДа обязательно учитывается специфика защищаемого объекта. Например, риски существенно выше при использовании виртуальных серверов. Из-за взаимосвязанной инфраструктуры завершенная атака на одного из клиентов может угрожать безопасности его «соседей».

Чаще всего используются автоматизированные средства безопасности, так как он позволяют добиться масштабирования на всех уровнях и автоматизировать контроль доступа. Применяются инструменты для фильтрации поступающего сетевого трафика на нескольких уровнях.

Нужно устранить «слепые зоны» на ступени виртуализации используемых ресурсов, в ином случае комплексные меры для ИБ могут оказаться малоэффективными.

Сегодня наиболее востребовано применение интегрированных и многоуровневых систем ИБ. В их состав входит макросегментация на уровне сетевого экрана и микросегментация используемых межсетевых экранов.

Следующий уровень защиты состоит в выявлении проблем внутри сегментов. В дата-центре осуществляется анализ и проверка динамики трафика, которая позволяет выявить возможные вредоносные активности (например, сканирование канала, DDoS-атаки, несанкционированное скачивание файлов и т.д.). Из-за больших объемов трафика сделать это «вручную» проблематично, поэтому применяются продвинутые алгоритмы. Эти инструменты позволяют распознать вредоносное ПО даже в шифрованном трафике.

Последний уровень – обеспечение защищенности устройств в локальной сети, то есть непосредственно самих серверов и виртуальных машин. Для этого используются устанавливаемые агенты для анализа операций ввода-вывода, изменения, удаления и любых других действий с файлами. Для анализа безопасности применяются алгоритмы Big Data, которые помогают выявить аномалии и нарушения логики.

Конечно, в ряде случаев удается обойтись без агентов установки. Многие способы защиты информации сегодня являются безагентными, то есть интегрируются в ОС на уровне гипервизора.

Для дата-центров, которые занимаются автоматизацией производственных процедур, связанных с высокой опасностью (например, на атомных электростанциях), представленных мер часто бывает недостаточно. Поэтому могут применяться дополнительные методики автоматизации.

Безопасность ЦОДов только за последние 10 лет шагнула далеко вперед: от простых методик физической защиты многие дата-центры перешли к использованию сложных интеллектуальных технологий. Конечно же, удалось решить не все возможные проблемы, но на сегодня уровень безопасности центров обработки информации достаточно высокий, поэтому подойдет даже для проектов с высокими требованиями к защите.

Важно понимать, что даже самые современные инструменты не могут отменить применения программных и технических средств, а также обучения персонала и организации ИБ внутри компании. И, конечно же, решить вопрос

защиты навсегда невозможно – это непрерывный и ежедневный процесс, который требует поиска новых уязвимостей и решения появляющихся проблем.

5.4.1. Понятие colocation. Критерии эффективности

Одной из популярных услуг центров обработки данных во всем мире является колокейшн. По статистике, эта услуга используется более чем 60% ИТ-компаний, а общий рынок достигает 54 млрд долларов. Что же такое colocation (колокейшн), для кого он подходит и как выбрать дата-центр?

Colocation – это услуга, представляющая собой размещение собственных серверов на специально оборудованной под эти цели площадке, а именно – в дата-центре (центре обработки данных). По сути – это физический хостинг, то есть вы самостоятельно устанавливаете сервер в выбранном ЦОД, выбираете программное и аппаратное обеспечение.

В дата-центрах созданы специальные условия для содержания серверов, которые сложно соблюдать в обычном офисе:

- присутствует необходимая вентиляция;
- есть независимый источник питания;
- поддерживается заданный температурный режим;
- отсутствует пыль и обеспечивается практически полная «стерильность»;
- установлена система видеонаблюдения и контроля;
- используется несколько интернет-провайдеров;
- присутствует квалифицированный обслуживающий персонал.

В результате размещения сервера в дата-центре обеспечивается его постоянное функционирование, присутствует возможность настройки и модернизации.

Преимущества colocation:

- защита сервера от перебоев питания;
- стабильная работа и безопасность всех данных;
- круглосуточная поддержка квалифицированного персонала;

– присутствие гарантий качества и заключенного с ЦОД соглашения;

– обеспечение высокой скорости интернет-соединения;

– оптимальная стоимость для малого и среднего бизнеса;

– возможность долгосрочной аренды площадей.

Предназначение услуги colocation.

В первую очередь такая услуга необходима тем компаниям, которые располагают собственным серверным оборудованием, но не имеют помещения для его установки. Чаще всего физический хостинг выбирается для ресурсоемких проектов. Обычно – это интернет-магазины, провайдеры и порталы. То есть это любые сайты, работающие с большим объемом трафика и крупными базами данных.

Услуга актуальна для компаний, старающихся оптимизировать свои расходы в период начинающегося кризиса. Самостоятельное содержание серверов требует постоянных расходов на аренду помещения, выполнение необходимого ремонта и обслуживания. Использование услуг дата-центра обходится дешевле, так как в стоимость аренды уже включены все расходы на работу персонала, видеонаблюдение и т.д. Обеспечивается более высокая скорость интернет-соединения, чем это было возможно в обычном офисе. То есть за меньшие деньги вы получаете весь спектр услуг профессионального уровня.

Выбор дата-центра для колокейшн.

Выбор ЦОД – ответственный шаг, который потребует внимания и займет много времени. Лучше всего определить основные параметры поиска, ответив на следующие вопросы:

1. Требуется ли физический или виртуальный сервер.

2. Какие задачи должен решать центр обработки информации.

3. Какая пропускная способность ЦОД вам требуется.

4. Оборудование какого типа планируется использовать.

5. Какие требования вы предъявляете к стабильности функционирования дата-центра и безопасности данных.

При подборе нужной площадки обязательно оцените ее расположение, уровень надежности и безопасности, мощностные ресурсы, охват связи, бесперебойность функционирования.

Категории центров обработки данных.

На основе надежности ЦОД делят на 4 уровня:

Tier I. Минимальный уровень надежности. Главная черта (и недостаток) таких площадок заключается в отказе от дублирующей инфраструктуры и коммуникаций. Ремонтные работы могут выполняться только при приостановке работы дата-центра. Именно по этой причине такие площадки сегодня мало распространены из-за своей непрактичности.

Tier II. Такие площадки также приостанавливают свою деятельность при необходимости проведения планового ремонта, однако имеют резервирование (дубли сетевого оборудования, системы кондиционирования и пр.). Значение отказоустойчивости достаточно большое и достигает 99,7%.

Tier III. Подобные центры обработки данных такого уровня функционируют без перебоев, даже во время запланированных работ. Располагаются в отдельном строении, присутствует дублирование основной техники и коммуникаций. Показатель отказоустойчивости равняется 99,98%.

Tier IV. ЦОД, обладающие максимальным коэффициентом отказоустойчивости – он достигает 99,99%. Простой в работе не более 15 минут в течение года.

Оптимальным для колокейшна являются ЦОД типа Tier II и Tier III. Они поддерживают высокий отказоустойчивость и предоставляют доступную аренду площадей.

6. АДМИНИСТРИРОВАНИЕ ЭЛЕМЕНТОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ СЕРВЕРОВ НА ОСНОВЕ ОПЕРАЦИОННЫХ СИСТЕМ LINUX

6.1. Основы обеспечения безопасности серверов на базе ОС Linux

6.1.1. Пользователи, права доступа и файловые системы

Первым шагом в защите Linux (да и любой другой системы семейства UNIX) является взятие под строгий контроль пользовательских и групповых учетных записей. Сюда входит принудительный выбор надежных паролей, предоставление доступа только к тем файлам, каталогам и интерпретаторам команд, которые действительно необходимы, и правильное управление учетной записью суперпользователя. В Linux есть множество утилит, позволяющих контролировать безопасность пользовательских учетных записей и групп. В этой главе будут описаны основные средства управления учетными записями, права доступа к файлам и каталогам, а также атрибуты файловых систем, имеющие отношение к контролю доступа.

6.1.2. Управление пользовательскими учетными данными

Linux – многопользовательская среда. Это означает, что к одному и тому же компьютеру, работающему под управлением Linux, могут одновременно обращаться множество пользователей. Каждому из них соответствует учетная запись. И каждая из них несет потенциальную угрозу безопасности системы. В процессе создания и управления учетными записями следует придерживаться определенных правил:

- выбор надежных паролей;
- обязательная проверка наличия пароля у пользователя;

- использование механизма устаревания паролей и файла скрытых паролей;
- ограничение доступа пользователей к файлам;
- использование подключаемых модулей аутентификации;

использование утилит *xlock* или *xscreensaver*.

6.1.3. Выбор надежных паролей

Выбор надежного пароля — не такая простая процедура для системного администратора, как может показаться на первый взгляд. Необходимо, чтобы пароль было трудно отгадать, но в то же время легко запомнить. Первое требование означает, что пароль не должен никак ассоциироваться с пользователем, т.е. он не должен представлять собой имя жены или ребенка, кличку собаки, номер автомобиля и т.п. Пароли, выбранные в нарушение этого правила, могут быть разгаданы человеком, который лично знаком с пользователем.

В качестве пароля нельзя выбирать слова из словаря. Программное обеспечение, предназначенное для перебора паролей, а особенно то, которое использует ресурсы графического ядра при их подборе, например, «hashcat» - легко «вычисляет» такие пароли, даже несмотря на то, что они хэшированы. Программа последовательно хэширует элементы словаря и сравнивает результаты с хэшированным паролем. При обнаружении совпадения пароль считается подобранным.

Один из способов выбора надежного пароля заключается в следующем. Сначала составляется легко запоминаемая фраза, например «The two foxes jumped over the brown fence». Далее можно пойти разными путями; будет описано два из них. Первый — просто взять начальные буквы каждого слова и образовать из них новое слово. Исключение составляет только слово «two», которое лучше заменить цифрой. В результате получается пароль «T2fjotbf». Второй вариант — взять вторую букву каждого слова, за исключением, опять-

таки, слова «two». Получится «h2ouvhrc». Можно придумать и другие, более изощренные схемы. Суть в том, чтобы придумать пароль, который трудно отгадать, но легко запомнить.

Другой подход к выбору надежных паролей заключается в их принудительном назначении. Можно написать скрипт-генератор паролей, воспользоваться существующим или самостоятельно придумывать пароли наподобие «i%8TQ#lr». Для администратора это самый простой способ контроля над паролями. Но пользователям чрезвычайно сложно будет запоминать подобного рода пароли и единственное, что им останется – это записать его на бумаге или в текстовый файл, что является серьезной уязвимостью и может привести к плачевным последствиям. Ситуация еще больше ухудшается, если используется механизм устаревания паролей, заставляющий пользователей периодически менять пароли.

Какие бы усилия ни предпринимались, даже самые надежные пароли оказываются бесполезными при наличии быстрых и недорогих компьютеров, утилит типа «hashcat» и анализаторов трафика. Если злоумышленник каким-то образом захватил список учетных записей и хешированных паролей, он может загрузить список в утилиту hashcat и почти наверняка получить на выходе удовлетворительный результат. Когда пользователь регистрируется в системе по сети, его имя и пароль часто передаются в открытом виде. В этом случае задача злоумышленника становится почти тривиальной. Ему нужно лишь воспользоваться анализатором трафика для перехвата всех передаваемых пакетов. В одном из них будет записан незашифрованный пароль. В учебном пособие рассматривается множество средств противодействия такого рода атакам, но в основе большинства из них лежит простое правило – бдительность системного администратора и пользователя.

Описанные выше атаки имеют высокие шансы на успех не в последнюю очередь из-за того, что регистрационные пароли Linux по умолчанию статичны. Имеется в виду, что один и тот же пароль многократно используется до тех пор, пока пользователь или администратор не поменяет его. Вместо статических

паролей можно перейти на одноразовые пароли (стандарт OTP). Но это довольно нетипичная ситуация, когда одноразовые пароли применяются в локальной сети организации. Гораздо привычнее, когда они используются для передачи по незащищенным или внешним каналам.

6.1.4. Обязательная проверка наличия пароля у пользователя

Категорически недопустимо наличие учетных записей без пароля. Если есть хотя бы одна такая запись, взлом системы будет лишь вопросом времени. Когда нужно предоставить кому-то гостевой доступ в систему, необходимо создать учетную запись с ограниченным сроком действия и назначьте ей пароль.

В Linux есть целый ряд учетных записей, которые заблокированы, т.е. они не имеют корректного пароля и, следовательно, не допускают регистрацию в системе. В листинге 6.1 приведено несколько записей из файла паролей (файл /etc/shadow не используется). Стоит обратить внимание на записи, у которых в поле пароля стоит звездочка. Через эти учетные записи может зарегистрироваться только суперпользователь с помощью команды su (или sudo). Каждая строка файла представляет собой отдельную запись. Запись — это группа полей, разделенных двоеточиями.

```
root:otlYOTgVSe. Bk:0:0 :root:/root:/bin/bash

bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mary:.Q17yU3a8bNrK:103:103:Mary
Smith:/home/mary:/bin/bash
```

Листинг 6.1. Фрагмент файла etc/passwd.

Учетные записи, не имеющие корректного пароля (символ * во втором поле), существуют для того, чтобы некоторые программы могли выполняться со специальными идентификаторами пользователя (UID). Зарегистрироваться в системе от имени такого пользователя нельзя.

Заблокированные учетные записи имеют еще одно применение. Если пользователь покидает организацию, но по каким-то причинам необходимо на время сохранить его среду – администратор блокирует учетную запись. Суперпользователь всегда сможет получить к ней доступ при необходимости. Наиболее часто взлому подвергаются длительное время не используемые учетные записи. Необходимо их блокировать или удалять.

Чтобы запретить доступ к заблокированной учетной записи, недостаточно применить описанный выше прием с паролем. Нужно также удалить файлы .rhosts, .netrc и .forward из начального каталога пользователя и задать в файле /etc/passwd фиктивный интерпретатор команд, например /dev/null или /bin/false. Последний является сценарием, выполняющим команду *exit*. Таким образом, даже если злоумышленник сможет зарегистрироваться в системе через заблокированную учетную запись, он будет немедленно выведен из системы.

6.1.5. Использование механизма устаревания паролей и файла скрытых паролей

Механизм устаревания паролей ограничивает срок, в течение которого пароль пользователя остается корректным. Этот механизм может быть реализован на общесистемном уровне посредством файла /etc/login.defs либо для отдельных пользователей посредством файла /etc/shadow. Последний называют файлом скрытых паролей. Для каждого пользователя он содержит запись, в которой указаны хешированный пароль и сведения о сроке его действия. По умолчанию этот файл может не существовать.

В Red Hat 7.2 файл скрытых паролей по умолчанию создастся в ходе инсталляции системы, что существенно повышает ее безопасность. Дело в том, что все хешированные пароли переносятся из файла /etc/passwd в файл /etc/shadow, доступ к которому разрешен только суперпользователю и только для чтения. Кроме того, появляется возможность задать предельный срок действия каждого пароля. Файл /etc/passwd открыт для чтения, а значит, и для копирования. Если файл скрытых паролей не используется, все хешированные пароли находятся в файле /etc/passwd, и любой пользователь может попытаться «взломать» его с помощью утилиты hashcat.

Чтобы создать файл скрытых паролей в Red Hat 7.2 (предполагается, что файл не был создан во время инсталляции), необходимо выполнить команду *pwconv* от имени суперпользователя. Она автоматически переместит в файл /etc/shadow все хешированные пароли (а также пароли вида *) и запишет во второе поле каждой записи файла /etc/passwd символ «x». Если нужно произвести обратное действие, необходимо выполнить команду *pwunconv* от имени суперпользователя. Разумеется, информация об устаревании паролей, хранящаяся в файле /etc/shadow, будет потеряна.

Механизм устаревания паролей отдельных пользователей включается только при наличии файла /etc/shadow. С помощью команды *chage* в этот файл можно добавлять сведения о пользовательских паролях. Фрагмент файла представлен в листинге 6.2.

```
root: otlY/YgV5e.Bk .-10640:0:99999:7: : :
bin:*:10640:0:99999:7: : :
daemon:*:10640 : 0 : 99999: 7 : : :
adm:*:10640:0:99999:7:: :
lp10640:0:99999: 7 : : :
sync:*:10640:0:99999:7: : :
joe:E/ulR7fLAQ06o:10640:0:99999:7:::
mary:VBtHXaJk31PD6:10640:7:30:3:45::
guestl:xdbt9JIxfCUvo:10640:0:99999:7: : 10640 :
```

Листинг 6.2. Фрагмент файла etc/shadow.

Подобно файлу passwd, файл shadow содержит одну строку (запись) для каждого пользователя. Каждая запись состоит из списка полей, разделенных двоеточиями. Всего существует 9 полей (табл. 6.1). Порядок записей в обоих файлах должен быть идентичным.

Таблица 6.1. Поля файла /etc/shadow[1]

Номер	**Команда для модификации**	**Описание**
1	useradd, usermod и userdel	Регистрационное имя пользователя; должно в точности соответствовать имени пользователя в файле /etc/passwd
2	passwd	Хешированный пароль длиной от 13 до 34 ASCII-символов; допустимые символы ~ a-z, A-Z, 0-9, '.', $ и /
3	passwd или chage -d	Дата последнего изменения пароля (число дней, прошедших с начала эпохи UNIX[2])
4	chage -m	Число дней, которое должно пройти с момента последнего изменения пароля, прежде чем его снова можно будет поменять
5	chage -M	Число дней, которое должно пройти с момента последнего изменения пароля, когда пользователь обязан будет сменить пароль
6	chage -W	Число дней до истечения срока действия пароля, когда выдается предупреждение
7	chage -I	Число дней по истечении срока действия пароля, когда учетная запись блокируется
8	chage -E	Число дней, которое должно пройти с начала эпохи UNIX, чтобы учетная запись устарела. В полночь последнего дня учетная запись будет заблокирована
9	–	Зарезервировано

[1] Не обязательно задействовать все параметры, перечисленные в таблице. Параметры, связанные с устареванием пароля, могут быть использованы по усмотрению администратора.

[2] Начало эпохи UNIX – 1 января 1970 года.

6.1.6. Ограничение доступа пользователей к файлам

Если необходимо создать гостевую учетную запись, необходимо назначить ей интерпретатор команд с ограниченными возможностями. В Red Hat 7.2 для этого есть специальная версия интерпретатора Korn (ksh), в которую введены следующие ограничения:

– команда *cd* недоступна;

– переменные среды *SHELL, ENV* и *PATH* нельзя изменить;

– разрешается выполнять только команды, которые расположены в каталогах, перечисленных в переменной среды *PATH*;

– операции переадресации ввода-вывода, приводящие к созданию новых файлов, запрещены.

Единственный способ вызова такого интерпретатора (в Red Hat 7.2) — воспользоваться командой */bin/ksh -r* или */bin/bash -r*. Это создает немалые неудобства, так как в файле /etc/passwd флаги командной строки не распознаются. Придется написать специальную программу, вызывающую интерпретатор с ограниченными возможностями (листинг 4.5). Сохраните программу в файле rksh. c, скомпилируйте се командой

$ gcc -o rksh rksh.c

скопируйте полученный исполняемый файл под именем /bin/rksh и сделайте его владельцем пользователя root (режим доступа — 755).

Владельцу гостевой учетной записи можно также запретить запись в начальный каталог. Для этого назначьте владельцем каталога специального пользователя, например noaccess, который не может зарегистрироваться в системе (нет корректного пароля, нет корректного интерпретатора команд, нет файлов . rhosts, . netre и . forward). Группа, которой принадлежит каталог, должна быть аналогичной. Учетную запись noaccess нужно создать специально для этих целей и ни для чего более. Задайте режим доступа к каталогу равным 755, и пользователь нс сможет осуществлять запись в каталог.

Создавать учетные записи с ограниченными возможностями можно и по-другому. Один из способов — воспользоваться командой chroot (меняет корневой каталог). Благодаря ей пользователь окажется в подкаталоге, который выглядит как корневой каталог, т.е. действия пользователя будут ограничены иерархией этого каталога. Существует также функция chroot (), которой можно пользоваться в программах. Сформированную таким способом среду необходимо тщательно протестировать.

6.2. Дополнительные механизмы обеспечения безопасности серверов на базе ос Linux

6.2.1. Использование функций системного обновления

Своевременно устанавливать обновления для ОС — несомненно положительное качество администратора. Разумеется, бывают случаи, когда обновление влечет за собой негативные последствия, но это происходит крайне редко. Данный процесс можно упростить, если использовать автообновления системы.

В разных дистрибутивах это делается по-своему:

– Debian/Ubuntu используют пакет unattended upgrades для автоматических обновлений;

– CentOS использует yum-cron;

– В дистрибутиве Fedora используется dnf-automatic.

Если сервер под критической нагрузкой, то следует использовать штатные средства.

Ubuntu/Debian:

```
sudo apt-get update
sudo apt-get upgrade
```

Fedora/Centos:

```
yum update
```

Обновление коснется только тех пакетов и приложений, которые не были установлены путем компилирования и получены как исполняемые файлы.

6.2.2. Системные пользователи с ограниченными правами

Подключение к серверу под учетной записью суперпользователя root небезопасно. Кроме того, рекомендуется сменить любого non-root пользователя, присутствующего в системе по умолчанию.

Изменение пароля производится командой:

```
passwd
```

Команда изменит пароль для того пользователя, от которого запускается. Если нужно изменить пароль для другого пользователя, следует выполнять команду следующим образом:

Для Ubuntu/Debian:

```
passwd <username>
```

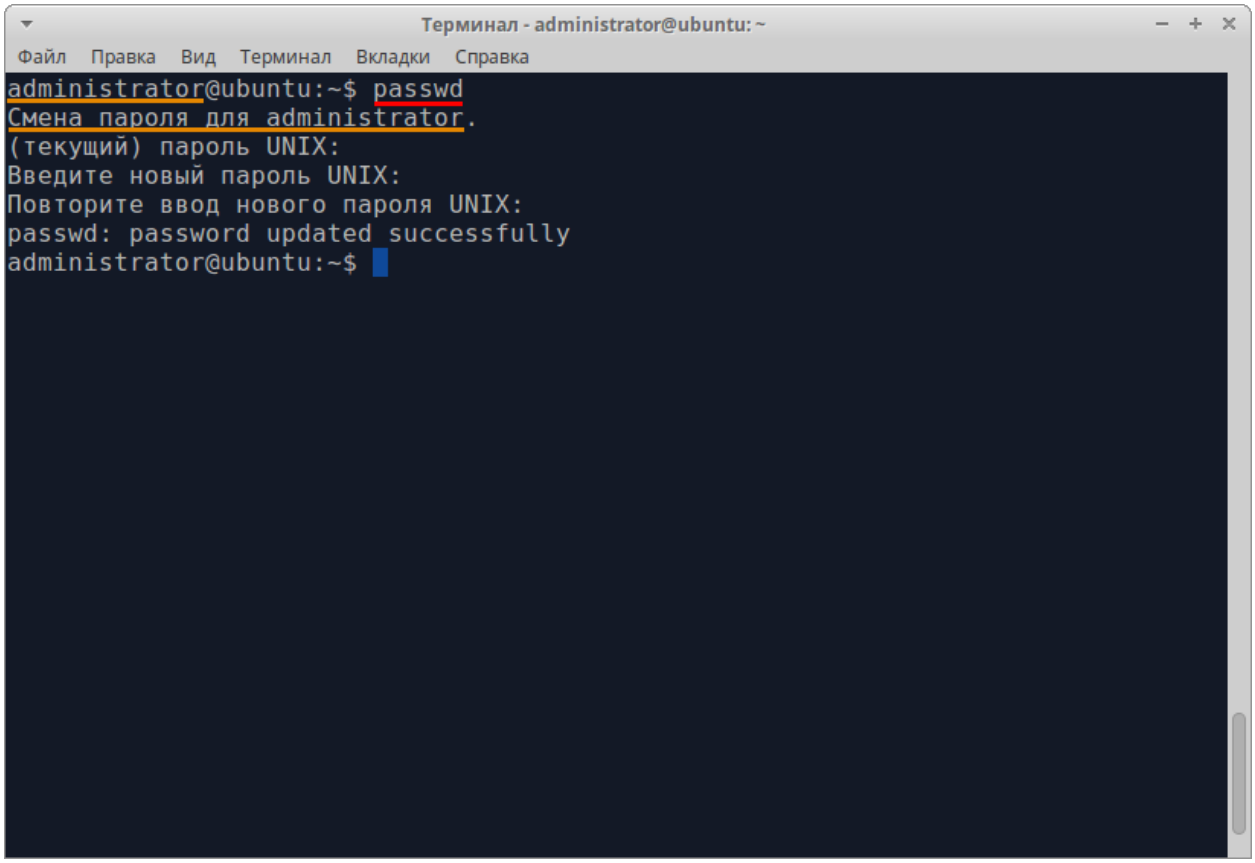

Рисунок 6.1. Смена пароля для Ubuntu/Debian

Для создания пользователя следует использовать команду:

```
adduser <user_name>
```

В ходе своей работы программа запросит пароль для учетной записи, сведения о пользователе, которые можно пропустить нажатием Enter.

По окончании заполнения данных программа запросит подтверждение корректности информации.

Следует пользователя правами администратора. Для этого добавим пользователя в группу sudo:

```
adduser <user_name> sudo
```

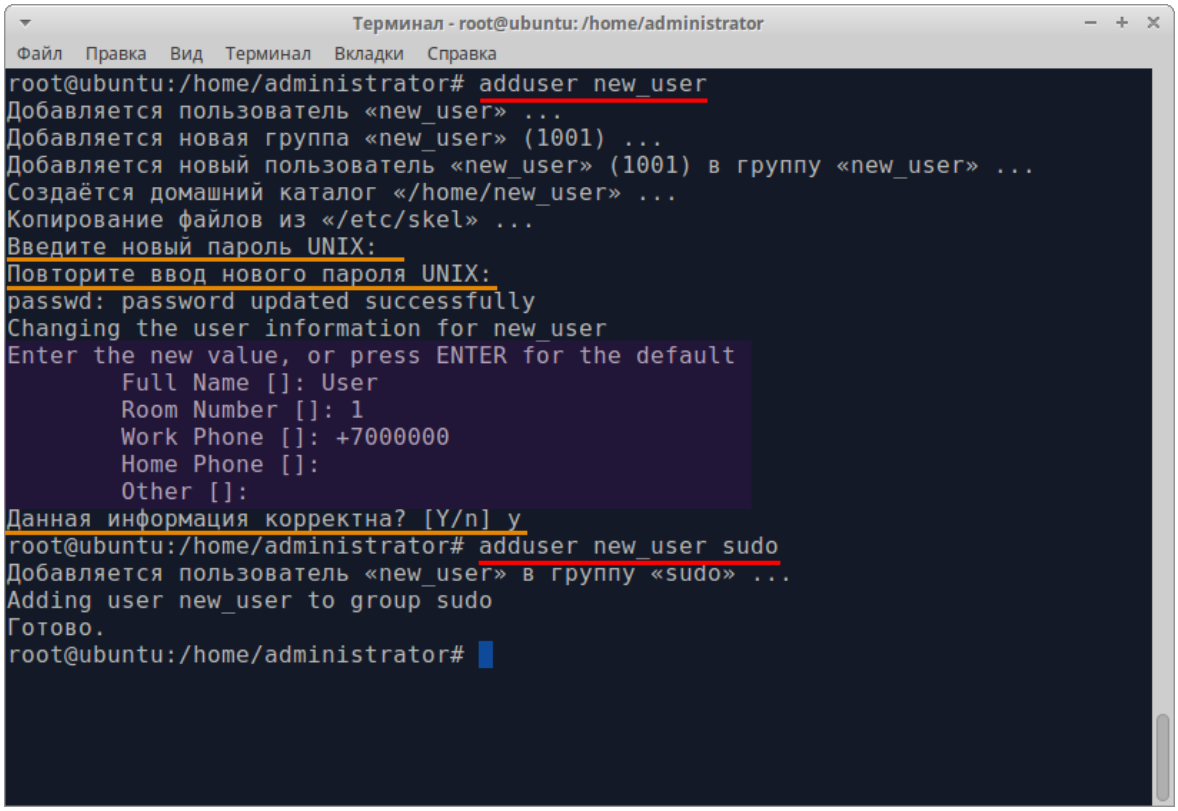

Рисунок 6.2. Добавление нового пользователя.

В случае с CentOS/Fedora пользователь создается иначе:

```
useradd <user_name> && passwd <user_name>
```

Добавить пользователя в группу wheel:

```
usermod -aG wheel <user_name>
```

6.2.3. Безопасное соединение по SSH

По умолчанию, доступ к серверу Linux осуществляется по паре логин-пароль на 22 TCP-порт. Согласно последним стандартам безопасности, рекомендуется поменять адрес порта сервера, а подключение производить по паре логин-ключ.

Изначально следует проверить генерировались ли ранее ключи для данной учетной записи:

```
ls ~/.ssh/id_rsa*
```

Если результат будет не пустым, то следует пропустить шаг создания ключа. Для создания ключа следует использовать команду:

```
ssh-keygen -t rsa
```

В ходе выполнения программы, может быть запрошена парольная фраза и ее подтверждение. Своего рода защита ключа паролем.

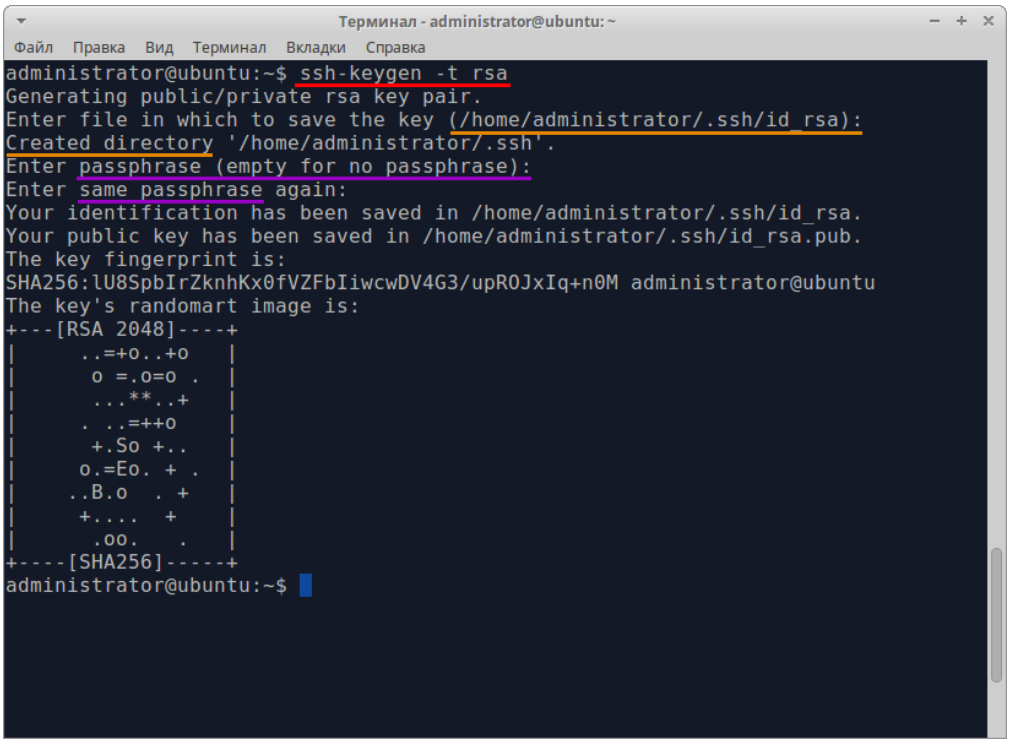

Рисунок 6.3. Генерация SSH ключей.

Для генерации ключа в Windows подходит программа PuTTY-Gen. После запуска программы, следует выбрать тип ключа RSA и кликнуть по кнопке Generate.

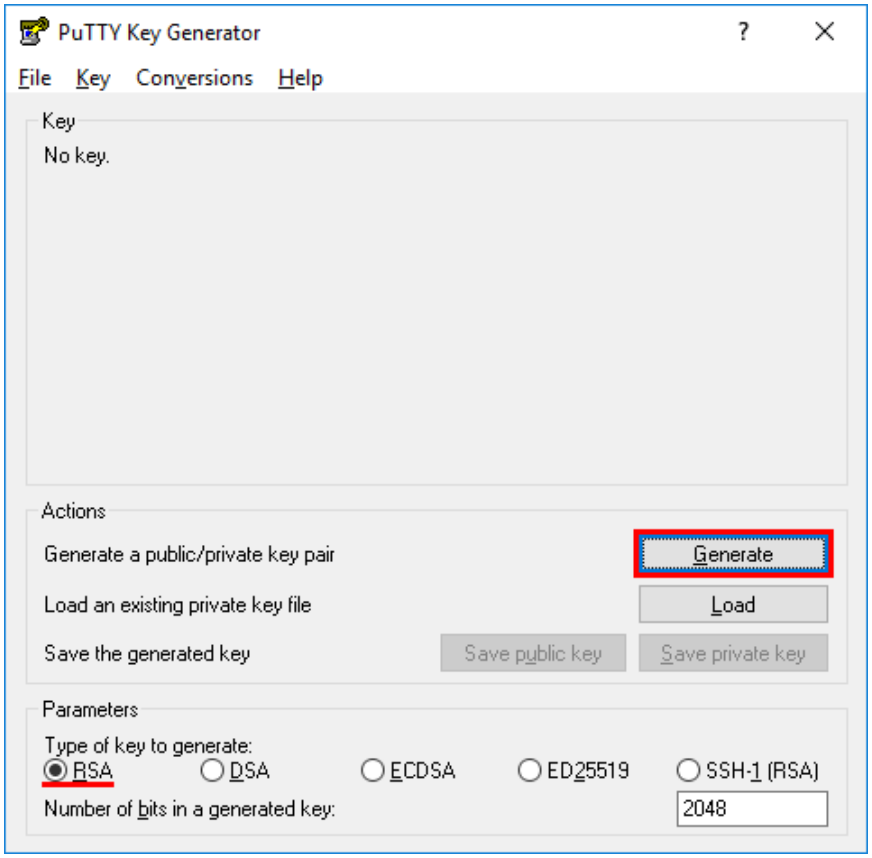

Рисунок 6.4. Генерация SSH ключей средствами Windows.

По окончании процедуры пользователю предоставляется следующая информация:

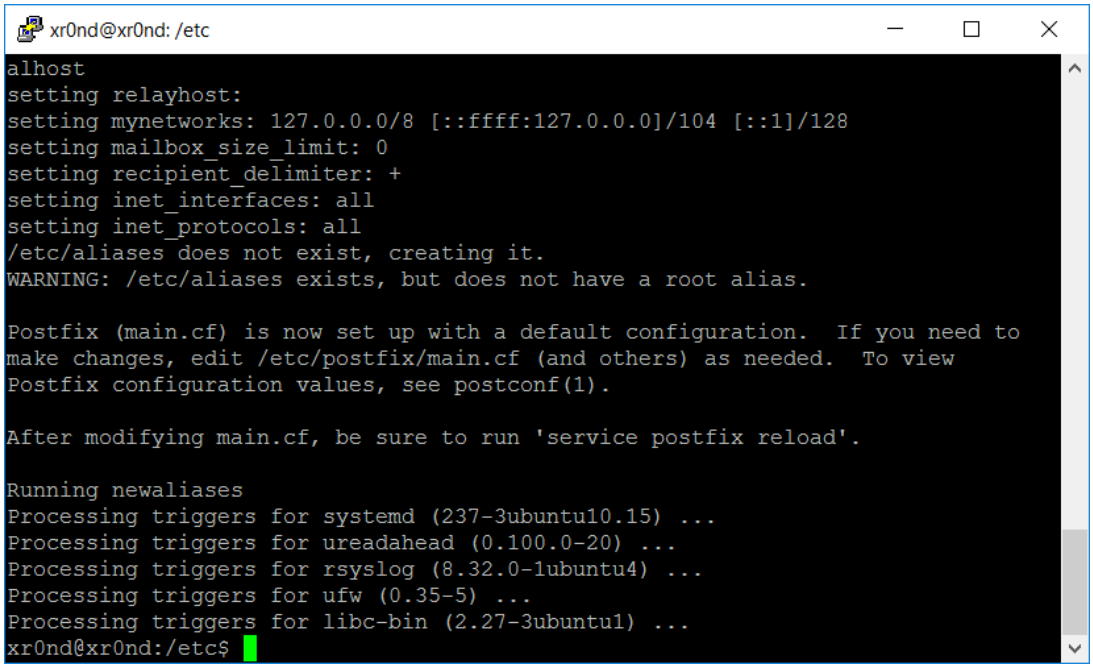

Рисунок 6.5. Вывод PuTTY-Gen.

Следует скопировать публичный ключ и сохранить приватный в файл, после чего добавить его на удаленный сервер.

Linux:

```
ssh-copy-id remoteuser@10.10.0.1
```

6.2.4. Настройка службы SSH

Следует запретить авторизацию от пользователя root. Ваша учетная запись должна быть в группе sudo или wheel (в зависимости от ОС) для того чтобы выполнять команды от суперпользователя, как это сделать указано выше. Для непосредственного выполнения команд следует перед командой использовать служебную команду sudo.

Например:

```
sudo reboot
```

Также для перехода в режим суперпользователя можно использовать одну из двух команд:

`su` либо `sudo su`

Во всех случаях система запросит пароль. Теперь следует отключить авторизацию под пользователем root. Необходимо открыть на редактирование файл sshd_config. В ***Debian/Ubuntu*** это выглядит так:

```
sudo nano /etc/ssh/sshd_config
```

Необходимо найти строчку *PermitRootLogin* и заменить его значение на no. Далее – необходимо запретить аутентификацию по паролю. Важно это сделать, если у вас уже успешно выполняется подключение по ключу.

Следует файл *sshd_config*:

`sudo nano /etc/ssh/sshd_config` найти строчку `PasswordAuthentication yes` и заменить значение на `PasswordAuthentication no`.

Строка может быть закомментирована (т.е. перед ней стоит символ «#»), в этом случае ее надо раскомментировать. Можно оставить авторизацию как по паролю, так и по ключу. По окончании настроек любого из пунктов, следует перезапустить сервер:

`sudo /etc/init.d/ssh restart` или `sudo service ssh restart` или `sudo systemctl restart sshd`.

6.2.5. Защита SSH-соединений с помощью Fail2Ban

Fail2Ban — приложение, которое позволяет блокировать SSH-подключения с определенного IP-адреса по достижении лимита. Разумно полагать, что если пользователь знает пароль к серверу, но ошибается при вводе, то достаточно будет 3-5 попыток. В случае с авторизацией по ключу — достаточно 1-2 попыток. В противном случае это брутфорс.

Fail2Ban способна осуществлять мониторинг и других протоколов, таких как HTTP, HTTPS, FTP и прочие. Используя настройки по умолчанию, осуществляется только мониторинг SSH.

Программа умеет противостоять различным атакм на все популярные *NIX–сервисы, такие как Apache, Nginx, ProFTPD, vsftpd, Exim, Postfix, named, и т.д.

Но в первую очередь Fail2ban известен благодаря защите SSH от перебора паролей.

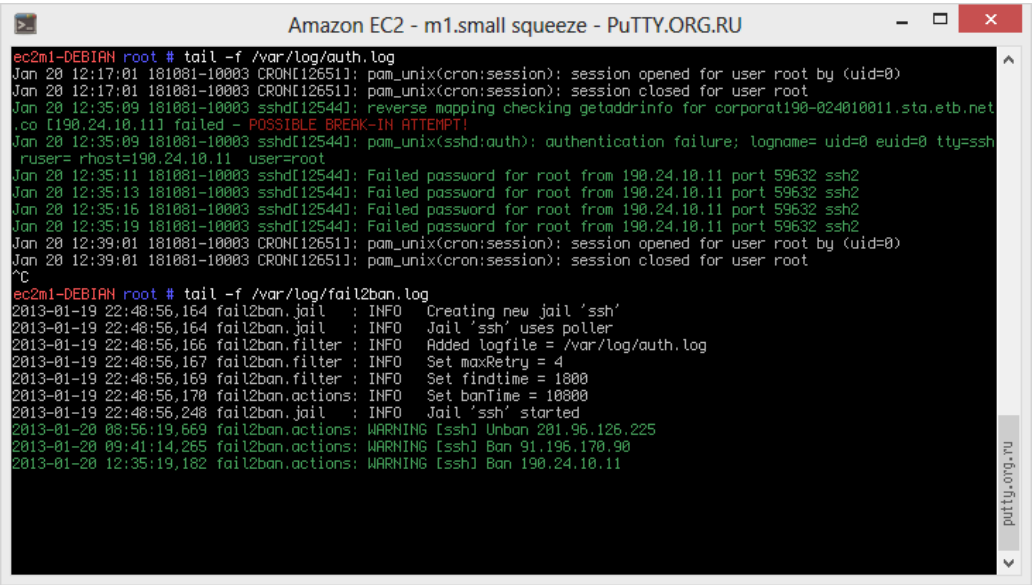

Рисунок 6.6. Статистика Fail2ban.

Готовые пакеты Fail2ban можно найти в официальных репозиториях всех популярных Linux дистрибутивов.

Установка Fail2ban на ***Debian/Ubuntu:***

```
apt-get install fail2ban
```

Установка Fail2ban на ***CentOS/Fedora/RHEL:***

```
yum install fail2ban
```

На данном этапе Fail2ban уже готов к работе, базовая защита SSH сервера от перебора паролей будет включена по умолчанию. У программы два основных файла конфигурации:

– `/etc/fail2ban/fail2ban.conf` — отвечает за настройки запуска процесса Fail2ban;

– `/etc/fail2ban/jail.conf` — содержит настройки защиты конкретных сервисов, в том числе sshd.

Файл `jail.conf` поделён на секции, так называемые «изоляторы» (jails), каждая секция отвечает за определённый сервис и тип атаки:

```
[DEFAULT]
ignoreip = 127.0.0.1/8
bantime  = 600
maxretry = 3
banaction = iptables-multiport
[ssh]
enabled  = true
port     = ssh
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 6
```

Листинг 6.3. Содержимое файла jail.conf

Параметры из секции [DEFAULT] применяются ко всем остальным секциям, если не будут переопределены. Секция [ssh] отвечает за защиту SSH от повторяющихся неудачных попыток авторизации на SSH–сервере, проще говоря, «brute–force».

Подробнее по каждому из основных параметров файла jail.conf:

– *ignoreip* — IP–адреса, которые не должны быть заблокированы. Можно задать список IP-адресов разделённых пробелами, маску подсети, или имя DNS–сервера.

– *bantime* — время бана в секундах, по истечении которого IP–адрес удаляется из списка заблокированных.

– *maxretry* — количество подозрительных совпадений, после которых применяется правило. В контексте [ssh] — это число неудавшихся попыток логина, после которых происходит блокировка.

– *enabled* — значение true указывает что данный jail активен, false выключает действие изолятора.

– *port* — указывает на каком порту или портах запущен целевой сервис. Стандартный порт SSH–сервера — 22, или его буквенное наименование — ssh.

– *filter* — имя фильтра с регулярными выражениями, по которым идёт поиск «подозрительных совпадений» в журналах сервиса. Фильтру sshd соответствует файл `/etc/fail2ban/filter.d/sshd.conf`.

– *logpath* — путь к файлу журнала, который программа Fail2ban будет обрабатывать с помощью заданного ранее фильтра. Вся история удачных и неудачных входов в систему, в том числе и по SSH, по умолчанию записывается в log–файл `/var/log/auth.log`.

Не рекомендуется оставлять параметр *ignoreip* со значением по умолчанию 127.0.0.1/8, это создаёт очевидную угрозу в многопользовательских системах — если злоумышленник получил доступ хотя–бы к одному shell–аккаунту, то он имеет возможность беспрепятственно запустить bruteforce–программу для атаки на root или других пользователей прямо с этого–же сервера.

Новая опция *findtime* — определяет длительность интервала в секундах, за которое событие должно повториться определённое количество раз, после чего санкции вступят в силу. Если специально не определить этот параметр, то будет установлено значение по умолчанию равное 600 (10 минут). Проблема в том, что ботнеты, участвующие в «медленном брутфорсе», умеют обманывать стандартное значение. Иначе говоря, при maxretry равным 6, атакующий может проверить 5 паролей, затем выждать 10 минут, проверить ещё 5 паролей, повторять это снова и снова, и его IP забанен не будет. В целом, это не угроза, но всё же лучше банить таких ботов.

Прежде чем вносить изменения следуя рекомендациям, отметим, что не стоит редактировать основной файл настроек jail.conf, для этого предусмотрены файлы с расширением *.local, которые автоматически подключаются и имеют высший приоритет.

```
nano /etc/fail2ban/jail.local
[DEFAULT]
## Постоянный IP-адрес.
## Если не переопределить ignoreip здесь,
## то стоит закомментировать этот параметр в jail.conf.
ignoreip = 57.66.158.131

[ssh]
## если в течении 1 часа:
findtime    = 3600
## произведено 6 неудачных попыток логина:
maxretry    = 6
## то банить IP на 24 часа:
bantime     = 86400
```

Листинг 6.4. Содержимое файла конфигурации.

Далее – следует перезапустить Fail2ban:

```
service fail2ban restart
tail /var/log/fail2ban.log
```

```
: INFO   Jail 'ssh' stopped
: INFO   Exiting Fail2ban
: INFO  Changed logging target to /var/log/fail2ban.log
for Fail2ban v0.8.6
: INFO   Creating new jail 'ssh'
: INFO   Jail 'ssh' uses poller
: INFO   Added logfile = /var/log/auth.log
: INFO   Set maxRetry = 6
: INFO   Set findtime = 3600
: INFO   Set banTime = 86400
: INFO   Jail 'ssh' started
```

Листинг 6.5. Содержимое .log файла

6.2.6. Настройка Firewall

Хорошей привычкой является использование файрволла. Это главный инструмент безопасности любого сервера и/или сети находящейся за ним. Фильтрация трафика позволяет избежать различного рода вторжения. Рекомендуется предоставлять доступ только к тем TCP/UDP-портам, которые на самом деле необходимы. По критичным портам либо закрывать доступ к ним, либо ограничивать доступ к ним — только с определенных IP-адресов.

IPTables — утилита командной строки, стандартный интерфейс управления работой межсетевого экрана netfilter в Linux. Для использования IPTables требуются права суперпользователя. Существуют и альтернативные решения UFW и ShoreWall для Debian/Ubuntu и FirewallD для CentOS и Fedora.

В данной лекции рассматривается настройка непосредственно IPTables.

Для просмотра действующих правил фильтрации используют следующие команды.

Ipv4:

```
sudo iptables -L
```

Ipv6:

```
sudo ip6tables -L
```

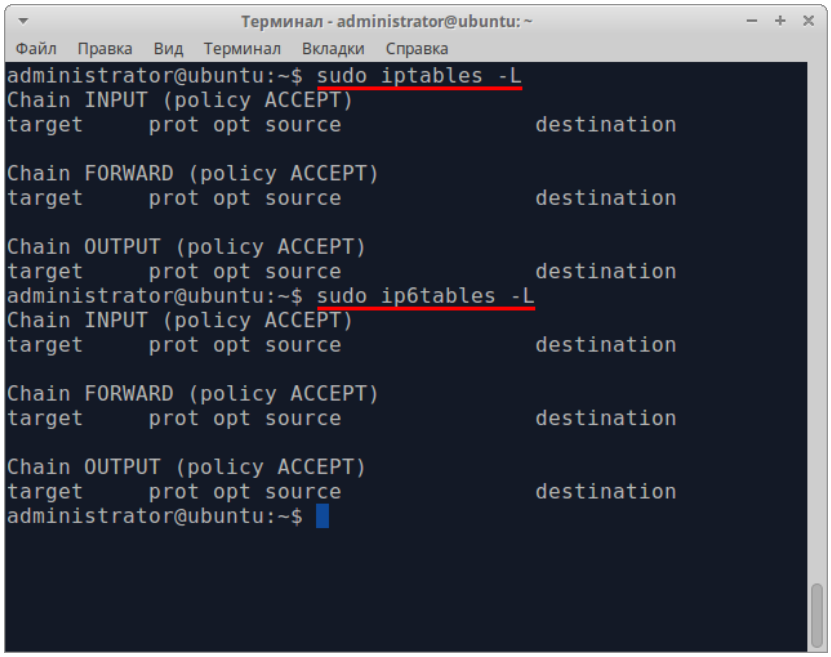

Рисунок 6.7. Вывод информации iptables

Это означает, что в режиме работы по умолчанию разрешен весь входящий, исходящий и пересылаемый трафик. Настройка межсетевого экрана и политика его работы зависит от работы сервисов, а также сервисов локальной сети (частный случай проброс порта для RDP).

Для IPv4 (файл /tmp/v4):

```
*filter
# Allow all loopback (lo0) traffic and reject traffic
# to localhost that does not originate from lo0
-A INPUT -i lo -j ACCEPT
-A INPUT ! -i lo -s 127.0.0.0/8 -j REJECT
# Allow ping.
-A INPUT -p icmp -m state --state NEW --icmp-type 8 -j ACCEPT
# Allow SSH connections.
-A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
# Allow HTTP and HTTPS connections from anywhere
# (the normal ports for web servers).
-A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
-A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT
# Allow inbound traffic from established connections.
# This includes ICMP error returns.
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# Log what was incoming but denied (optional but useful).
-A   INPUT   -m   limit   --limit   5/min   -j   LOG   --log-prefix
"iptables_INPUT_denied: " --log-level 7
# Reject all other inbound.
-A INPUT -j REJECT
# Log any traffic that was sent to you
# for forwarding (optional but useful).
-A   FORWARD   -m   limit   --limit   5/min   -j   LOG   --log-prefix
"iptables_FORWARD_denied: " --log-level 7
# Reject all traffic forwarding.
-A FORWARD -j REJECT
COMMIT
```

Листинг 6.6. Настройка iptables для IPv4

```
*filter
-A INPUT -i lo -j ACCEPT
-A INPUT ! -i lo -s ::1/128 -j REJECT
-A INPUT -p icmpv6 -j ACCEPT
-A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
-A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A   INPUT   -m   limit   --limit   5/min   -j   LOG   --log-prefix
"ip6tables_INPUT_denied: " --log-level 7
-A INPUT -j REJECT
-A   FORWARD   -m   limit   --limit   5/min   -j   LOG   --log-prefix
"ip6tables_FORWARD_denied: " --log-level 7
-A FORWARD -j REJECT
COMMIT
```

Листинг 6.7. Настройка iptables для IPv6

Для ***Arch Linux***:

Требуется создать файлы `/etc/iptables/iptables.rules` и `/etc/iptables/ip6tables.rules`. Далее - выставить правила из примеров выше (/tmp/v4 и /tmp/v6) в созданные файлы соответственно. После следует импортировать эти правила для применения iptables:

sudo iptables-restore < /etc/iptables/iptables.rules

sudo ip6tables-restore < /etc/iptables/ip6tables.rules

В Arch Linux, по умолчанию, iptables на запущен, его следует включить:

sudo systemctl start iptables && sudo systemctl start ip6tables

sudo systemctl enable iptables && sudo systemctl enable ip6tables

Для ***CentOS / Fedora:***

В данных дистрибутивах, применяемые правила хранятся в файлах `/etc/sysconfig/iptables` и `/etc/sysconfig/ip6tables`. Для управления правилами межсетевого экрана предлагается использование FirewallD, вместо непосредственного администрирования iptables. Как и в случае с Arch Linux, требуется создать файлы `/tmp/v4` и `/tmp/v6`.

Импорт и сохранение правил:

```
sudo iptables-restore < /tmp/v4
sudo ip6tables-restore < /tmp/v6
sudo service iptables save
sudo service ip6tables save
```

Для ***Debian/Ubuntu:***

В дистрибутивах Debian и Ubuntu правила можно писать как вручную, так и использовать утилиту UFW.

Импорт правил из файлов:

```
sudo iptables-restore < /tmp/v4
sudo ip6tables-restore < /tmp/v6
```

Для автоматизации и упрощения загрузки правил iptables при запуске сервера можно использовать пакет *iptables-persistent*.

sudo apt-get install iptables-persistent

sudo iptables -vL

sudo ip6tables -vL

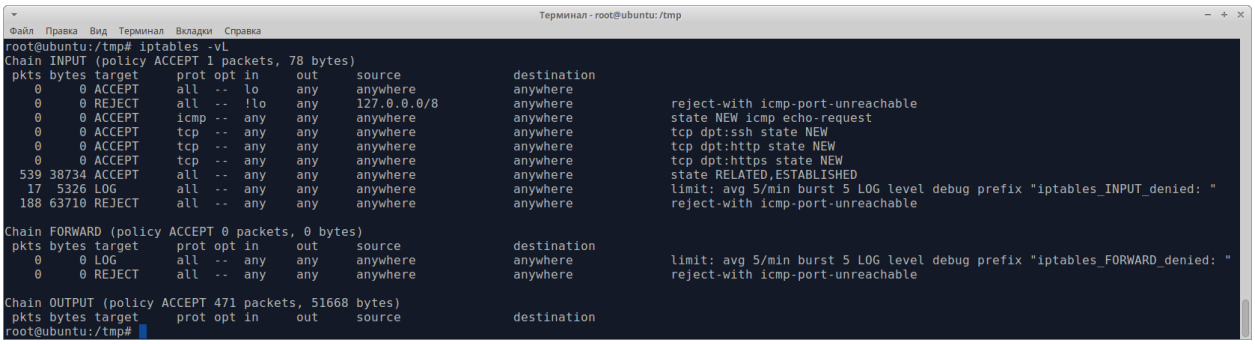

Рисунок 6.8. Результат iptables

После перезапуска, следует проверить правила. Правила должны присутствовать в том же количестве. Логика работы iptables такова, что правила работают последовательно от первого к последнему. По этой причине невозможно добавить правила привычными командами. Для добавления правил в этом случае используется:

```
iptables -I
ip6tables -I
```

Добавляемые правила должны располагаться в определенной последовательности с учетом других правил в цепочке. Для отображения нумерованного списка существует команда:

```
sudo iptables -L -line-numbers
```

Например, необходимо добавить новое разрешающее правило для соединений на порт 8080, к существующим ранее:

```
sudo iptables -I INPUT 9 -p tcp --dport 8080 -j ACCEPT
```

Замена правил выполняется ключом «-R»:

```
iptables -R
```

Удаление ранее добавленных правил:

```
sudo iptables -D INPUT 9
```

Будет удалено правило, в котором было разрешено подключение к 8080 порту. Применяемые правила не применяются автоматически.

6.2.7. Базы данных

Не менее важным является защита данных, находящихся в некой СУБД. Рассмотрим на примере MariaDB. После успешной установки необходимо выполнить одну команду:

```
sudo mysql_secure_installation
```

После чего, программа задаст несколько вопросов касаемых безопасности:

Таблица 6.2. Перечень вопросов MardiaDB

Change the root password? [Y/n]	Изменить пароль пользователя root?
Remove anonymous users? [Y/n]	Удалить анонимных пользователей?
Disallow root login remotely? [Y/n]	Запретить удаленное подключение от имени root?
Remove test database and access to it? [Y/n]	Удалить базу данных test и доступ к ней?
Reload privilege tables now? [Y/n]	Перезагрузить таблицу привилегий сейчас?

Результатом установки будет:

```
administrator@ubuntu:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.
```

Рисунок 6.9. Вывод результатов установки MariaDB

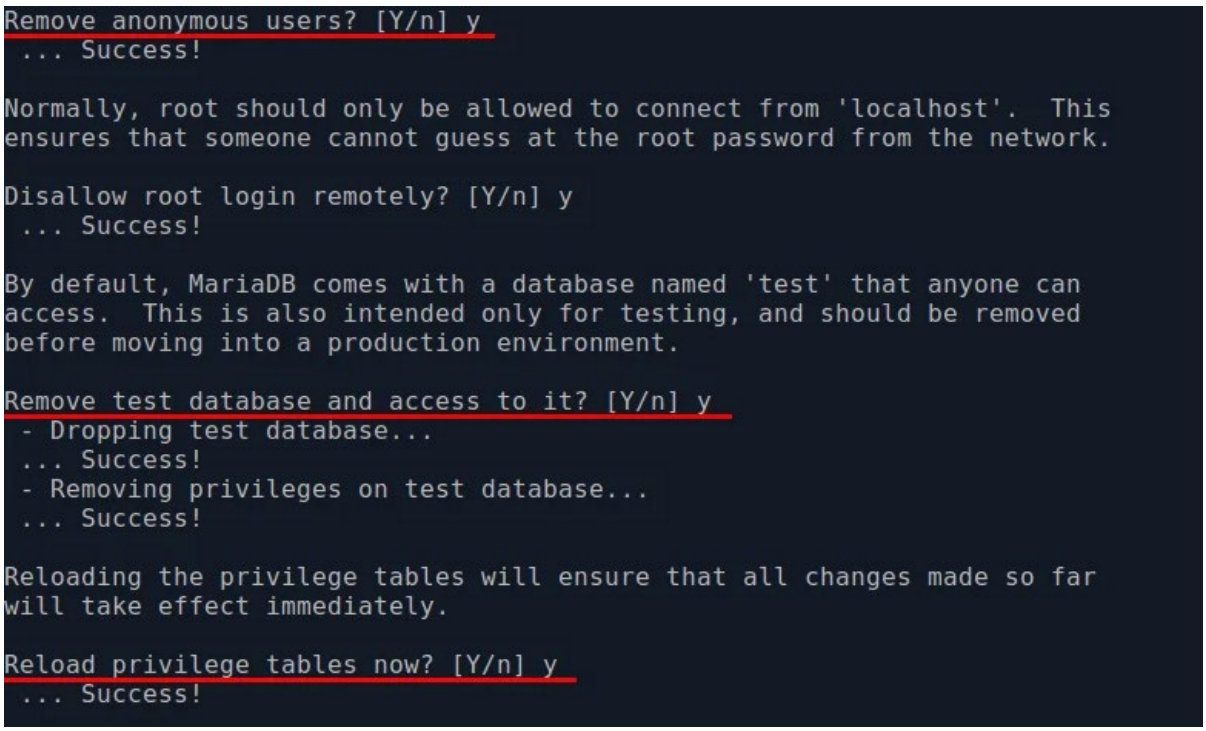

Рисунок 6.10. Вывод результатов установки MariaDB

Также не рекомендуется выполнять соединения от имени пользователя root. Лучше создать одного пользователя и с ограниченными правами. Для сайта будет достаточно следующих прав для выполнения запросов вида:

SELECT — выборка из базы

UPDATE — Обновление записей

INSERT — добавление новых записей.

DELETE — удаление записей (иногда, но лучше не использовать).

Не рекомендуется наделять правами:

ALTER — изменение структуры таблиц

DROP — удаление баз данных и таблиц базы

Вполне разумным является и то, чтобы один пользователь был для одной базы данных.

6.2.8. Общие требования к учетным данным

Для защиты сервера и баз данных следует использовать грамотно созданные имя пользователя и пароль.

Так как подбор имен и паролей (брутфорс) происходит по словарю, логично было бы использовать такое имя пользователя, которое с наименьшей вероятностью окажется в словаре. Например, xd11rn и подобные. Не стоит использовать слишком короткие имена пользователей. Главное, потом не забыть имя пользователя.

К паролям есть ряд общих требований:

- не использовать пароли короче 10 символов;
- использовать буквы верхнего и нижнего регистра, а также цифры;
- использовать специальные символы, но только там, где это возможно.

6.3. Инструментарий серверов на базе ОС Linux

6.3.1. Регулярные выражения в командной оболочке Bash

Регулярные выражения (RegEx или RegExp) — это набор символов, которые определяют шаблон поиска. Их можно использовать для выполнения операций «Поиск» или «Поиск и замена», а также для проверки условий, таких как политика паролей, ввод номера телефона и т.д.

Пример регулярного выражения:

```
/t[aeiou]l/
```

Это регулярное выражение будет искать слово, начинающееся с буквы ‘t’, содержащее любую из букв ‘a e i o u’ в середине и оканчивающееся буквой ‘l’. Это может быть ‘tel’, ‘tal’ или ’til’. Совпадение может быть отдельным словом или частью другого слова, например ’tilt’, ‘brutal’ или ‘telephone’.

В общем виде синтаксис команды ‘grep’ выглядит следующим образом:

```
$ grep поисковый_запрос_regex расположение_файла
```

Стоить понять назначение специальных символов, известных как метасимволы. Они помогают создавать более сложные поисковые выражения:

Таблица 6.3. Метасимволы регулярных выражений

Символ	Описание
.	будет соответствовать любому символу;
[ ]	будет соответствовать диапазону символов;
[^]	будет соответствовать всем символам, кроме указанных в фигурных скобках;
*	будет соответствовать любому количеству символов, предшествующих звездочке, в том числе нулю;
+	будет соответствовать одному или нескольким из стоящих перед ним выражений;
?	будет соответствовать нулю или одному из стоящих перед ним выражений;
{n}	будет соответствовать ‘n’ повторениям предшествующих выражений;
{n,}	будет соответствовать не менее ‘n’ повторениям предшествующих выражений;
{n m}	будет соответствовать не менее ‘n’ и не более ‘m’ повторениям предшествующих выражений;
{,m}	будет соответствовать не более или равному ‘m’ повторениям предшествующих выражений;
	является escape-символом (символом экранирования), используемым, когда нужно включить один из метасимволов.

. (точка)

Используется для соответствия любому символу, который встречается в поисковом запросе. Например, можно использовать точку как:

```
$ grep "d.g" file1
```

Это регулярное выражение означает, что мы ищем слово, которое начинается с ‘d’, оканчивается на ‘g’ и может содержать один любой символ в середине файла с именем ‘file1’. Точно так же мы можем использовать символ точки любое количество раз для нашего шаблона поиска, например:

```
T.....h
```

Этот поисковый термин будет искать слово, которое начинается с ‘T’, оканчивается на ‘h’ и может содержать любые шесть символов в середине.

[ ]

Квадратные скобки используются для определения диапазона символов. Например, когда нужно искать один из перечисленных символов, а не любой символ, как в случае с точкой:

```
$ grep "N[oen]n" file2
```

Здесь мы ищем слово, которое начинается с ‘N’, оканчивается на ‘n’ и может иметь только ‘o’, ‘e’ или ‘n’ в середине. В квадратных скобках можно использовать любое количество символов. Мы также можем определить диапазоны, такие как ‘a-e’ или ‘1-18’, как список совпадающих символов в квадратных скобках.

[^]

Это похоже на оператор отрицания для регулярных выражений. Использование [^] означает, что поиск будет включать в себя все символы, кроме тех, которые указаны в квадратных скобках. Например:

```
$ grep "St[^1-9]d" file3
```

Это означает, что у нас могут быть все слова, которые начинаются с ‘St’, оканчиваются буквой ‘d’ и не содержат цифр от 1 до 9.

До сих пор мы использовали примеры регулярных выражений, которые ищут только один символ. Но что делать в иных случаях? Допустим, если

требуется найти все слова, которые начинаются или оканчиваются символом или могут содержать любое количество символов в середине. С этой задачей справляются так называемые метасимволы-квантификаторы, определяющие сколько раз может встречаться предшествующее выражение: + * & ?

{n}, {n m}, {n, } или { ,m} также являются примерами других квантификаторов, которые мы можем использовать в терминах регулярных выражений.

**** (звездочка)***

На следующем примере показано любое количество вхождений буквы ‘k’, включая их отсутствие:

```
$ grep "lak*" file4
```

Это означает, что у нас может быть совпадение с ‘lake’ или ‘la’ или ‘lakkkkk’.

+ (плюс)

Следующий шаблон требует, чтобы хотя бы одно вхождение буквы ‘k’ в строке совпадало:

```
$ grep "lak+" file5
```

Здесь буква ‘k’ должна появляться хотя бы один раз, поэтому наши результаты могут быть ‘lake’ или ‘lakkkkk’, но не ‘la’.

? (знак вопроса)

В следующем шаблоне результатом будет строка bb или bab:

```
$ grep "ba?b" file6
```

С заданным квантификатором ‘?’ мы можем иметь одно вхождение символа или ни одного.

Предположим, у нас есть регулярное выражение:

```
$ grep "S.*l" file7
```

И мы получаем результаты ‘Small’, ‘Silly’, и ещё ‘Susan is a little to play ball’. Но почему мы получили ‘Susan is a little to play ball’, ведь мы искали только слова, а не полное предложение?

Все дело в том, что это предложение удовлетворяет нашим критериям поиска: оно начинается с буквы ‘S’, имеет любое количество символов в середине и заканчивается буквой ‘l’. Итак, что мы можем сделать, чтобы исправить наше регулярное выражение, чтобы в качестве выходных данных мы получали только слова вместо целых предложений.

Для этого в регулярное выражение нужно добавить квантификатор ‘?’:

```
$ grep "S.*?l" file7
```

или символ экранирования

Символ » используется, когда необходимо включить символ, который является метасимволом или имеет специальное значение для регулярного выражения. Например, требуется найти все слова, заканчивающиеся точкой. Для этого можем использовать выражение:

```
$ grep "S.*?." file8
```

Оно будет искать и сопоставлять все слова, которые заканчиваются точкой.

6.3.2. Способы запуска команды через фоновый режим Linux

Фоновый процесс – это любые команды, работа которых была запланирована юзером в принудительном порядке. Такие действия не будут отображаться на мониторе и не заметны рядовому пользователю. Яркий пример процесса – сервер Apache, обслуживающий веб-страницу, а также скрипты оболочек ПО и команды.

Работу многих команд, задач и скриптов возможно перестроить под фоновый процесс одним добавлением символа «&». Амперсанд следует поставить в командную строку в самый конец функции. После чего он быстро перенесет скрипты и освободит немного свободного места.

Те скрипты, которые будут работать через фоновый режим, называются заданиями. Синтаксис для их выполнения будет иметь такой вид:

`command & script-name & /path/to/command arg1 arg2 & command-1 | command-2 arg1 & command-1 | command-2 -arg1 -arg2 >/path/to/output &`

Рассмотрим простой пример перевода некоторой команды ls:

```
$ ls *.py > output.txt &
```

Для перевода последующего поискового скрипта нужно будет добавить амперсанд после всего выражения:

```
find . -iname "*.mp3" > myplaylist.txt &
```

Чтобы найти команды и скрипты, которые были переведены в фоновый режим, понадобится команда: `jobs`

```
[1]- Running find / -iname "*.c" 2> /dev/null > /tmp/output.txt &
[2]+ Running grep -R "hostNamed" / 2> /dev/null > /tmp/grep.txt &
```

Листинг 6.8. Пример работы утилиты jobs

В этом выражении [1] и [2] являются идентификаторами задания. Для их дальнейшего отображения через идентификатор процесса нужно будет ввести -l: `jobs –l`

```
[1]- 7307 Running find / -iname "*.c" 2> /dev/null > /tmp/output.txt &
[2]+ 7324 Running grep -R "hostNamed" / 2> /dev/null > /tmp/grep.txt &
```

Листинг 6.9. Пример работы утилиты jobs с ключом -l

Если нужно показать только идентификатор процесса, следует написать: `jobs -p`

Если администратору понадобилось постепенно или принудительно завершить фоновый процесс одной из задач, следует воспользоваться командой «`kill`»:

```
kill PID
kill -15 PID
kill -9 PID
killall process-Name-Here
killall -15 process-Name-Here
killall -9 process-Name-Here
```

Листинг 6.10. Пример работы утилиты kill

Для перемещения фонового процесса на место приоритетного используется: `%JOB-ID` либо `fg JOB-ID`

Поначалу выводятся все текущие фоновые процессы при помощи синтаксиса: `jobs -l`

```
[1]- 7307 Running find / -iname "*.c" 2> /dev/null > /tmp/output.txt &
[2]+ 7324 Running grep -R "hostNamed" / 2> /dev/null > /tmp/grep.txt &
```

Листинг 6.11. Вывод о перемещении процесса

Перевод задания со вторым идентификатором осуществится благодаря команде: `%2` либо `fg 2`

Как можно вывести данные:

```
grep -R "hostNamed" / 2> /dev/null > /tmp/grep.txt
```

Если потребуется приоритетные задания сделать снова фоновыми, достаточно будет удерживать «CTRL» и одновременно нажать «Z». Все работающие приоритетные задачи будут остановлены на некоторое время. Возвращение к фоновому режиму будет проводиться так: `%2 &` либо `bg`. Таким образом «grep» вернется к фоновому режиму.

Таблица 6.4 – Наиболее полезные команды при работе с процессами

Описание	**Команда**
Покажет все выполняемые задания, а также их идентификаторы.	*jobs* *jobs -l* *ps aux*
Переведет команду или скрипт для работы в приоритетном режиме.	*command &* */path/to/command &* */path/to/script arg1 &*
Переведет фоновое задание для работы в приоритетном режиме.	*fg n* *%n*
Переведет команду обратно к фоновому режиму, не запрашивая его отмену.	*bg n* *%n &*

Под «n» подразумевается идентификационный номер заданий, он выводится при выполнении команды «jobs».

6.3.3. Монтирование диска и создание разделов в Linux

Установленного в системе жесткого диска может быть недостаточно и иногда встает вопрос о добавлении дополнительного носителя. Рассмотрим как смонтировать диск на Virtual Private Server (VPS), Virtual Dedicated Server (VDS) и Standalone Server.

Перед началом работы следует убедиться существует ли диск в системе. Иногда наличие устройства в BIOS может быть недостаточным. Доступные накопители проверяем командой: `sudo fdisk -l`

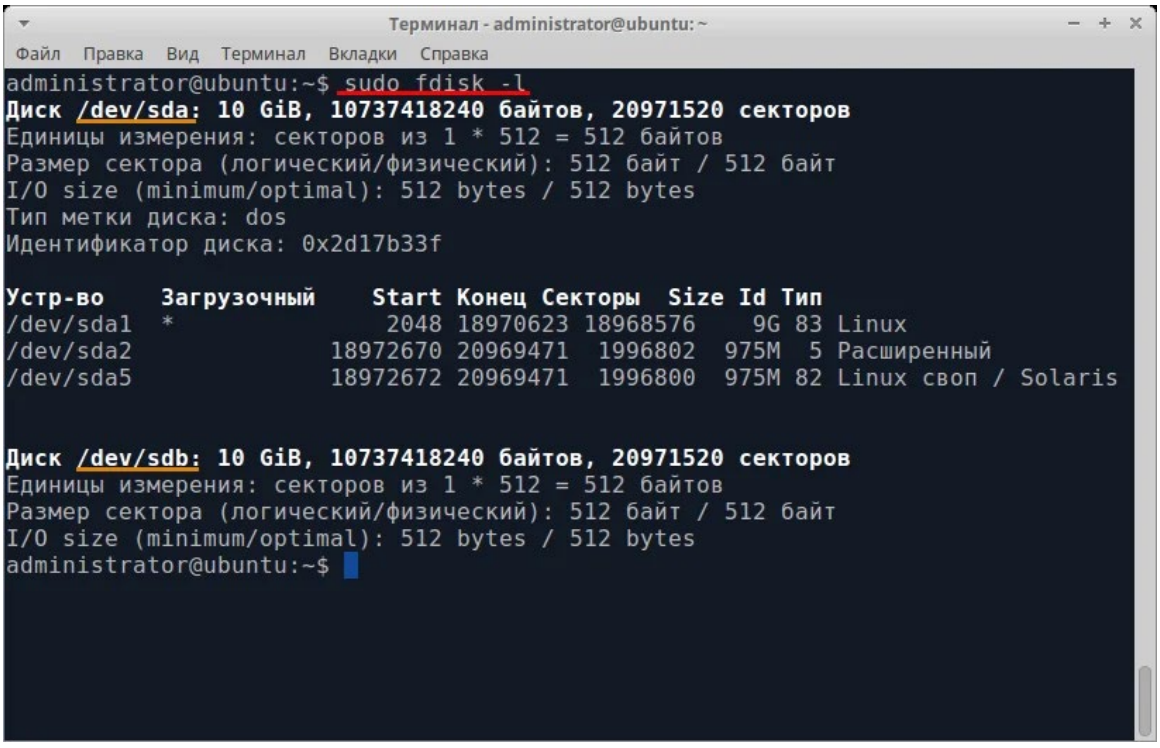

Рисунок 6.11. Вывод списка накопителей

В нашем случае в системе присутствуют 2 диска — sda и sdb, оба по 10 Гб, первый диск системный. Работать будем со вторым диском (sda). Важно

помнить, что Linux системы можно повредить неверно указав диск, например системный. Следует тщательно проверять и перепроверять выбор носителя. На помощь приходит тот факт, что новый диск не имеет файловой системы и разделов.

Для разметки диска запустить утилиту fdisk с указанием пути до диска: `fdisk /dev/sdb`.

При нажатии “m” и подтверждением ввода клавишей Enter, программа предоставит страницу доступных команд:

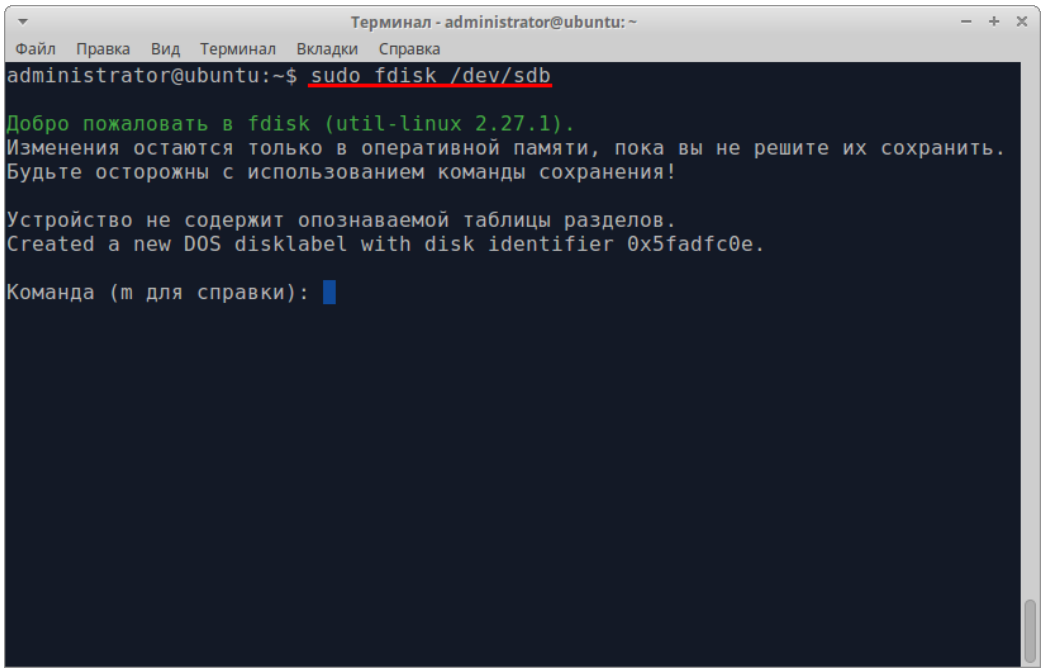

Рисунок 6.12. Вывод справки

```
Справка:
DOS (MBR)
a toggle a bootable flag
b edit nested BSD disklabel
c переключить флаг совместимости с DOS
 Общие
d удалить раздел
F list free unpartitioned space
l список известных типов разделов
n добавить новый раздел
p показать таблицу разделов
t изменить тип раздела
v проверить таблицу разделов
i print information about a partition
 Разное
m показать это меню
u изменить единицы отображения/ввода
x дополнительные функции (только для экспертов)
 Script
I load disk layout from sfdisk script file
O dump disk layout to sfdisk script file
 Сохранить и выйти
w сохранить таблицу на диск и выйти
q выйти без сохранения изменений
 Создать новую метку
g создать новую пустую таблицу разделов GPT
G создать новую пустую таблицу разделов SGI (IRIX)
o создать новую пустую таблицу разделов DOS
s создать новую пустую таблицу разделов Sun
```

Листинг 6.12. Справочная информация

Так как мы будем создавать простой раздел (не загрузочный), будем использовать все дисковое пространство, нажимаем клавишу “n” и Enter.

Система спросит о типе раздела — выбираем p (primary) — первичный. Номер раздела — 1. На вопрос о первом и последнем секторе отвечаем нажатием клавиши Enter. Если вы желаете создать несколько логических дисков, то в ответе на последний сектор следует указать размер в кило-, мега-, гига-, тера-, петабайтах. Соответственно, если раздел нужен размером в 2 гигабайт, то указываем 2G. Сохраняем изменения нажатием “w” и подтверждаем выбор с помощью клавиши Ввод.

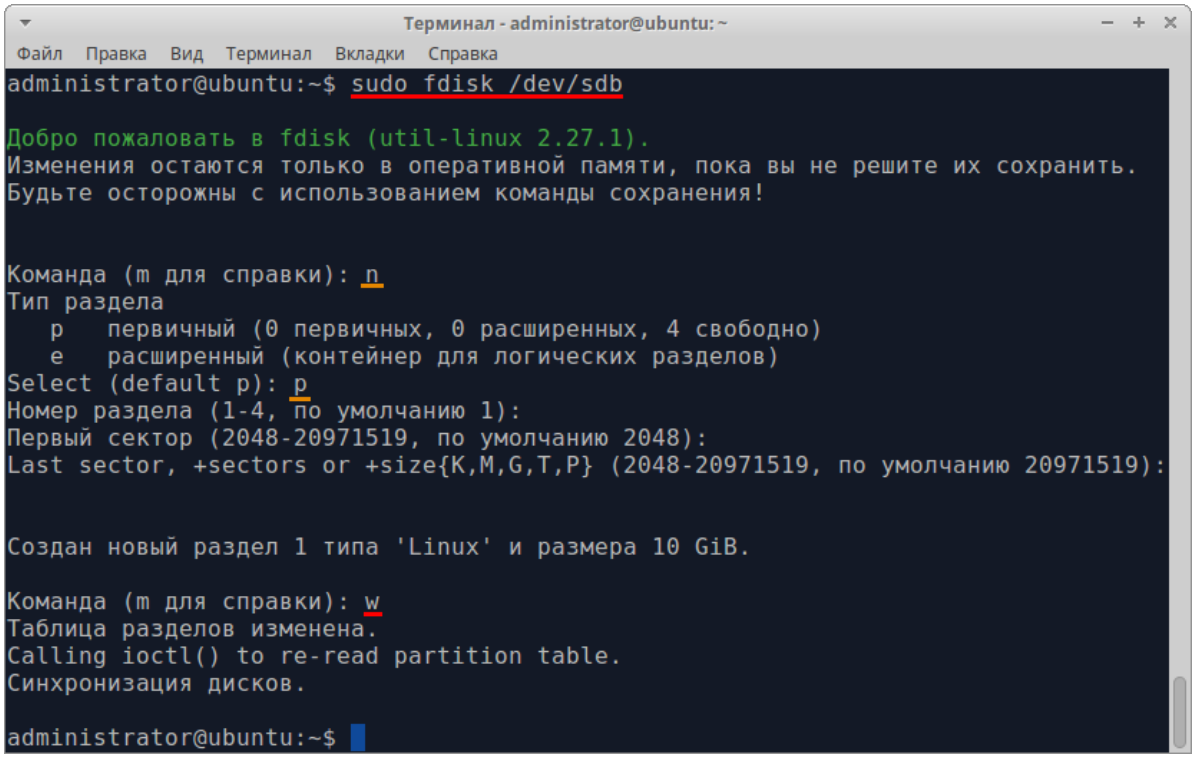

Рисунок 6.13. Вывод результатов

После выполнения операции описанной выше, в системе будет создано устройство `/dev/sdb1` — по своей сути это и есть раздел на диске. Современный Linux предлагает на выбор несколько вариантов файловых систем. Создание файловой системы происходит выполнением команды mkfs с указанием ключей, либо выполнением одной из программ:

- `mkfs.bfs`
- `mkfs.btrfs`
- `mkfs.cramfs`
- `mkfs.ext2`
- `mkfs.ext3`
- `mkfs.ext4`
- `mkfs.ext4dev`
- `mkfs.fat`
- `mkfs.minix`

- `mkfs.msdos`
- `mkfs.ntfs`
- `mkfs.vfat`
- `mkfs.xfs`

Форматирование выполняется командой:

```
sudo mkfs.ext4 /dev/sdb1
```

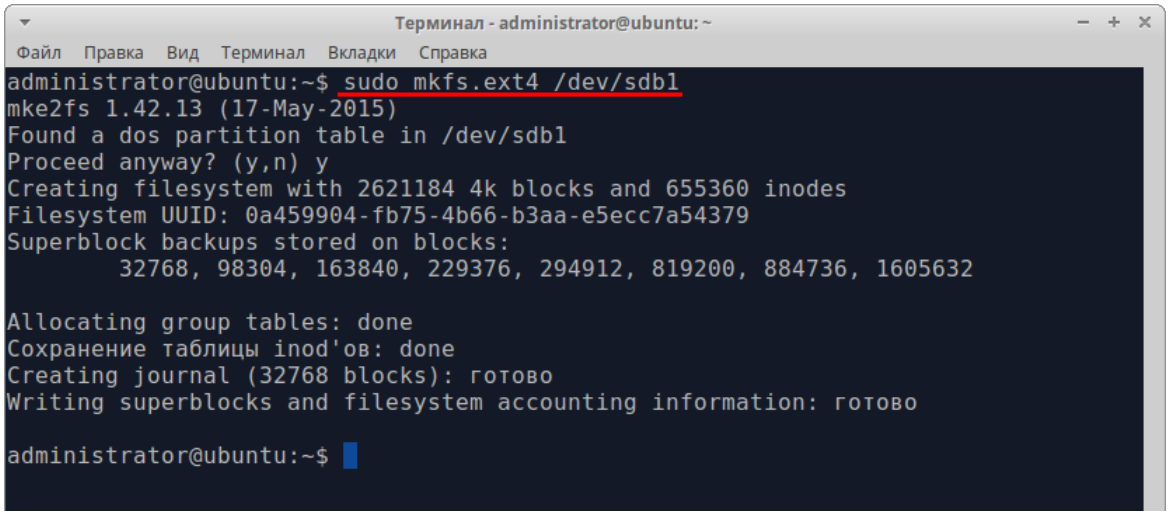

Рисунок 6.14. Вывод результатов форматирования

Диск готов к работе. Остается только смонтировать его. Особенность ОС Linux в том, что существует возможность монтировать диск (либо иное блочное устройство) в любую из директорий, главное чтобы директория была пустой. Создаем каталог в директории /mnt:

```
sudo mkdir /mnt/1
```

Изменяем права доступа к каталогу. Только root и и только чтение и запись.

```
sudo chmod -R 660 /mnt/1
```

Собственно, монтируем:

```
sudo mount /dev/sdb1 /mnt/1
```

Для монтирования диска автоматически при загрузке системы, редактируем файл `/etc/fstab`. Открываем любым текстовым редактором, например nano:

```
sudo nano /etc/fstab
```

В самый конец файла вставляем строку:

```
/dev/sdb1 /mnt/1 ext4 defaults 0 0
```

Сохраняем файл.

6.3.4. Настройка планировщика Cron в Linux

Планировщик заданий является по праву одним из важных компонентов системы. По умолчанию, в дистрибутивах Linux, а также FreeBSD, используется планировщик заданий Cron. Настроить cron можно несколькими способами. Самый простой и быстрый — указать задание руками в файл `/etc/crontab`. Проблема в том, что это может сделать только пользователь root.

Структура файла `/etc/crontab` представляет собой последовательность установок даты и времени, имени пользователя от которого выполняется задача и явное указание исполняемого файла.

`# m h dom mon dow user command`

– `#` — строка закомментирована, т.е. не выполняется планировщиком

– `m` — минуты. Диапазон значений 0-59

– `h` — часы. Диапазон значений 0-23.

– `dom – day of month` — день месяца (число). Диапазон значений 1-31

– `mon` — месяц. Диапазон значений 1-12

– `dow – day of week` — день недели. указывается числом, где 0 — воскресенье, 6 — суббота.

– `user` — пользователь от которого выполняется задача.

– `command` — выполняемая задача.

Значения минут, часов, дней месяца, дней недели, а также месяца могут быть * (звездочкой) принимая все доступные значения. Также можно использовать список параметров, разделенных запятой.

Простым пользователям доступна системная утилита crontab. Для гибкости использования используются параметры:

Таблица 6.5. Параметры утилиты Cron

Команда	**Описание**
crontab -e	Используется для редактирования или создания файла расписания для текущего пользователя
crontab -l	Вывод существующих задач в расписании пользователя
crontab -r	Удаление файла расписания текущего пользователя
crontab -u username	Работа с расписаниями указанных пользователей. Выполняется только от пользователя root

Добавим задание. Для этого в консоли пишем команду: `crontab -e`. Запустится редактор файла используемый по умолчанию (Ubuntu — nano/vi, FreeBSD — ee), в нашем случае mcedit из пакета Midnight Commander.

Синтаксис команд идентичен как и для файла `/etc/crontab`, за исключением отсутствия параметра имени пользователя. Это значит, что после сохранения файла, все задачи будут выполняться от текущего пользователя.

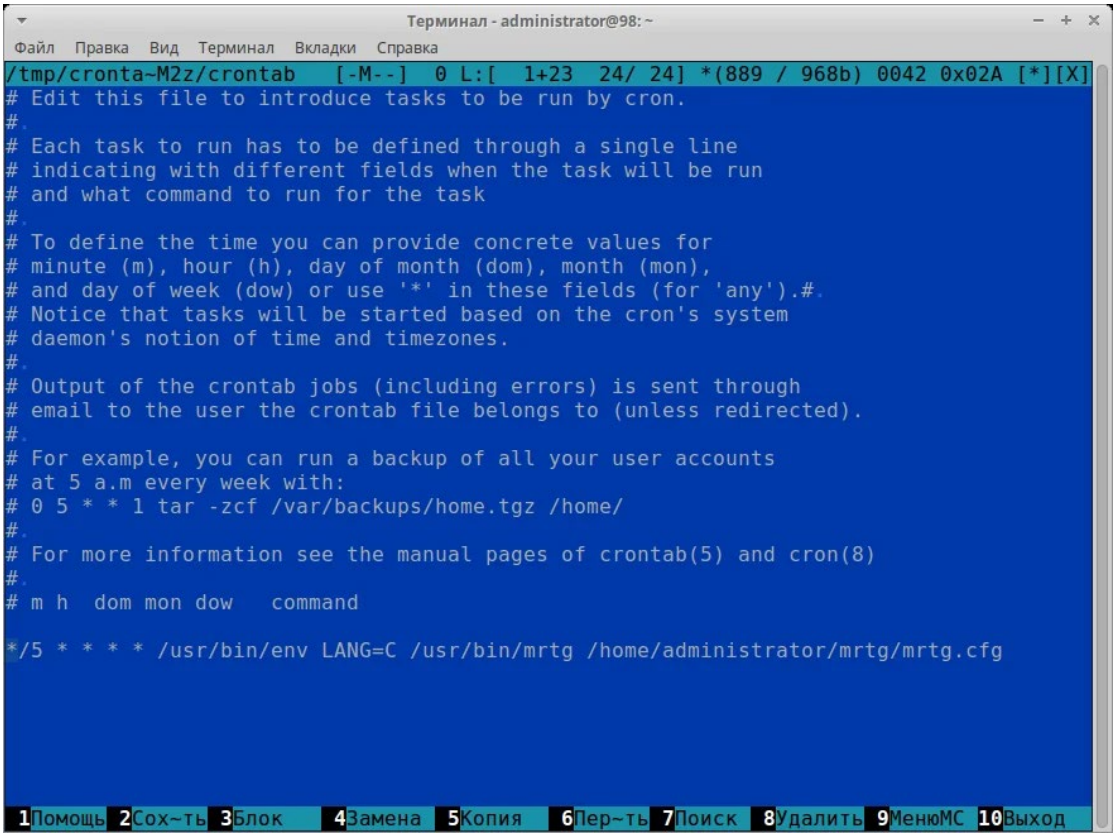

Рисунок 6.15 – Файл планировщика задач Cron

Сохраняем файл, выходим из редактора и проверяем список заданий `crontab -l`. Проверим как работает удаление файла заданий — `crontab -r`.

Программа не спрашивает уверены ли вы в удалении файла заданий. Если нужно удалить конкретное задание, следует использовать `crontab -e`.

Для администратора важно, чтобы не все пользователи могли пользоваться планировщиком. Для этих целей можно использовать файлы `/etc/cron.allow` и `/etc/cron.deny` (в некоторых дистрибутивах эти файлы должны размещаться в /etc/cton.d). По умолчанию, использование cron разрешено всем пользователям.

Выполнение задачи каждые 5 минут от пользователя root:

```
*/5 * * * * root /root/scripts/script1.sh
```

В полдень и полночь:

```
0 */12 * * * root /root/scripts/backup.sh
```

Перезагрузка сервера в 0:00 в первый день месяца (исключительно как пример):

```
0 0 1 * * root reboot
```

Выполнение скрипта с понедельника по пятницу в полночь:

```
0 0 * * 1-5 root /root/script/backup_db.sh
```

Неправильный пример:

```
* 0 * * 1-5 root /root/script/backup_db.sh
```

Ошибка в том, что скрипт будет запускаться с понедельника по пятницу каждую минуту с 0:00 по 0:59. В заданиях назначаемых через crontab -e (т.е. от текущего пользователя) не указывается параметр имени пользователя, в случае примеров — root. Все пути до исполняемых файлов следует писать абсолютными. Скрипты исполняются из корневой директории, а следовательно, при написании скрипта следует указывать рабочий каталог, либо абсолютные пути до целевых файлов (если в ходе работы скрипта генерируются файлы-отчета и им подобные).

6.3.5. Резервное копирование Linux-серверов

Создание резервных копий – одна из неотъемлемых задач системного администратора. Рассмотрим, как осуществить процедуру с помощью встроенных утилит dd и tar на семействах ОС Linux.

Утилита dd является одной из первых в своем направлении. У нее отсутствуют некоторые опции, которые доступны на современных аналогах, но она удобна и проста в использовании. К тому же, исходный файл бэкапа читается другими программами без проблем.

Запускаем утилиту с заданными параметрами:

```
dd if=xxx of=yyy bs=8M conv=sync,noerror
```

Синтаксис следующий. XXX – имя диска, с которого производится резервное копирование, а YYY – наименование файла бэкапа.

Операнд bs задает размер кэша жесткого диска. Установим его на 8 Мб, чтобы копирование происходило быстрее. Параметр conv указывает на то, что копирование файлов происходит побитно с пропуском ошибок чтения.

Команда tar в основном используется для создания архивов, но ее используют и для бэкапов:

```
tar -cvpzf name1.tar.gz --exclude=name2 --one-file-system folder1
```

– `name1` – произвольное имя для новой резервной копии;

– `exclude=name2` означает, что из архива исключаем все папки и файлы с таким именем;

– `one file system` сообщает архиватору о том, что используем только одну файловую систему;

– `folder1` – имя директории, резервная копия которой создается командой tar.

Чтобы выполнить восстановление из резервной копии, используем следующий синтаксис:

```
dd if=ZZZ of=QQQ bs=8M conv=sync,noerror
```

Расшифровка следующая. ZZZ – имя бэкапа с указанием пути до него. Например, /mnt/backup/21082019.img, QQQ – диск, на который восстанавливается копия.

Архиватор использует немного другой набор команд:

```
sudo tar -xvpzf XXX -C YYY --numeric-owner
```

– `XXX` – полное наименование архива;

– `YYY` – директория, в которой будет перемещена разархивированная информация.

Последний ключ (—numeric-owner) означает, что пользователи тоже будут восстановлены, но не по учетному имени, а по числовому значению.

6.3.6. Управление системными процессами Linux через Top

Операционные системы семейства Linux для каждой задачи или операции запускают отдельный системный процесс. Если во время работы происходит сбой службы или появляется ошибка, то необходимо принимать меры. Рассмотрим один из инструментов работы с процессами в ОС семейства Linux. С помощью Top процессам можно задавать приоритеты, просматривать полный список, а также завершать. Запуск осуществляется вводом команды в терминале: `top`. Приложение интерактивное, т.е. изменения происходят в режиме реального времени и отображаются на экране пользователя.

Верхняя часть утилиты отображает сведения о технических характеристиках компьютера (сервера), общем количестве сервисов, а также об объеме оперативной памяти и файла подкачки:

```
top - 18:43:13 up 22 min,  1 user,  load average: 0,00, 0,00, 0,02
Tasks: 223 total,   1 running, 187 sleeping,   0 stopped,   0 zombie
%Cpu(s):  9,4 us,  2,0 sy,  0,0 ni, 88,6 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
КиБ Mem :  2038860 total,   503428 free,   919456 used,   615976 buff/cache
КиБ Swap:   728520 total,   728520 free,        0 used.   955892 avail Mem
```

Рисунок 6.16 – Расшифровака top

Далее следует информация о процессах:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1215	inferno	20	0	408064	69076	38668	S	6,0	3,4	0:02.38	Xorg
1702	inferno	20	0	690532	39268	30540	S	2,6	1,9	0:00.32	gnome-flash+
1482	inferno	9	-11	1435176	13464	9884	S	0,7	0,7	0:00.11	pulseaudio
1994	inferno	20	0	867900	38432	29220	S	0,7	1,9	0:01.73	gnome-termi+
1544	inferno	20	0	1142244	27168	21520	S	0,3	1,3	0:00.15	gsd-media-k+
1691	inferno	20	0	501336	34048	26792	S	0,3	1,7	0:00.41	metacity
1695	inferno	20	0	822188	48436	36844	S	0,3	2,4	0:00.49	gnome-panel
1738	inferno	20	0	377776	10008	8432	S	0,3	0,5	0:00.11	ibus-daemon
2014	**inferno**	**20**	**0**	**51380**	**4012**	**3364**	**R**	**0,3**	**0,2**	**0:06.70**	**top**
1	root	20	0	159932	9304	6752	S	0,0	0,5	0:01.23	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp

Рисунок 6.17. Информация о процессах top

Каждая колонка отображает свое значение:

1. PID – идентификатор сервиса, который не повторяется. Он позволяет однозначно идентифицировать процесс.

2. USER – учетное имя, под которым запущена служба.

3. PR расшифровывается как приоритет расписания. Назначается на уровне ядра операционной системы.

4. NI означает nice. В терминологии Linux чем выше цифра, тем меньше системных ресурсов выделяется. Отрицательное значение показывает максимальный приоритет сервису.

5. VIRT – объем виртуальной памяти, который выделяется под конкретный процесс. Обозначает общий ресурс, т.е. в него входит информация, страницы памяти, коды ошибок и т.д.

6. RES – это часть VIRT, которая не помещена в раздел подкачки. По сути это физическое значение оперативной памяти, используемое данной службой.

7. SHR расшифровывается как совместная оперативная память. Она распределяется между другими процессами.

8. S – статус. Каждой службе присваивается свой код в зависимости от принятых кодов:

- **D** = бесперебойный сон
- **I** = простой
- **R** = запущен
- **S** = спит
- **T** = остановлен сигналом управления работой
- **t** = остановлен отладчиком во время трассировки
- **Z** = зомби

Рисунок 6.18. Расшифровка кодов S

9. %CPU и %MEM – потребляемый ресурс процессора и памяти. Рассчитывается в процентном соотношении от общего времени.

10. TIME+ отображает время, которое служба использовала с момента своего запуска. Рассчитывается от общего времени работы процессора.

11. COMMAND проводит соответствие между процессом и программой, которая его запустила.

Навигация в терминале осуществляется с помощью клавиатуры. Стрелки вверх и вниз перемещаются между строками, а вправо и влево отвечают за горизонтальную навигацию.

Клавиша I убирает процессы, которые не используются:

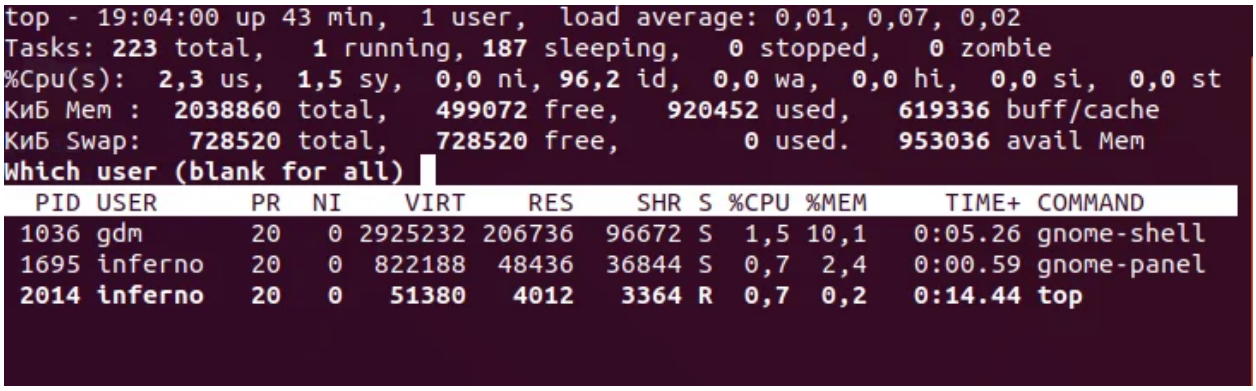

Рисунок 6.19. Убрать неактивные процессы

Кнопка Z включает/выключает цветовой режим:

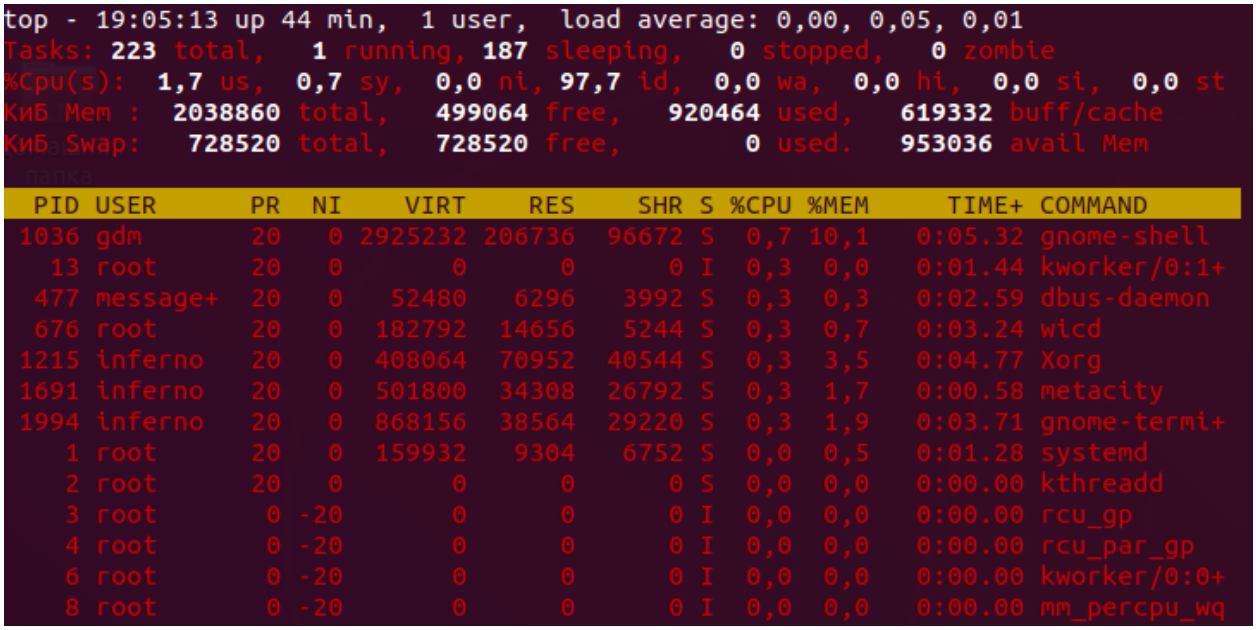

Рисунок 6.20. Активация цвета

Если требуется отобразить полный путь к приложению, которое запустило выбранный процесс, нажимаем клавишу C.

Нажатие клавиши K приведет к завершению выбранного сервиса. На экране отобразится дополнительная строка, в которой указываем идентификатор (PID):

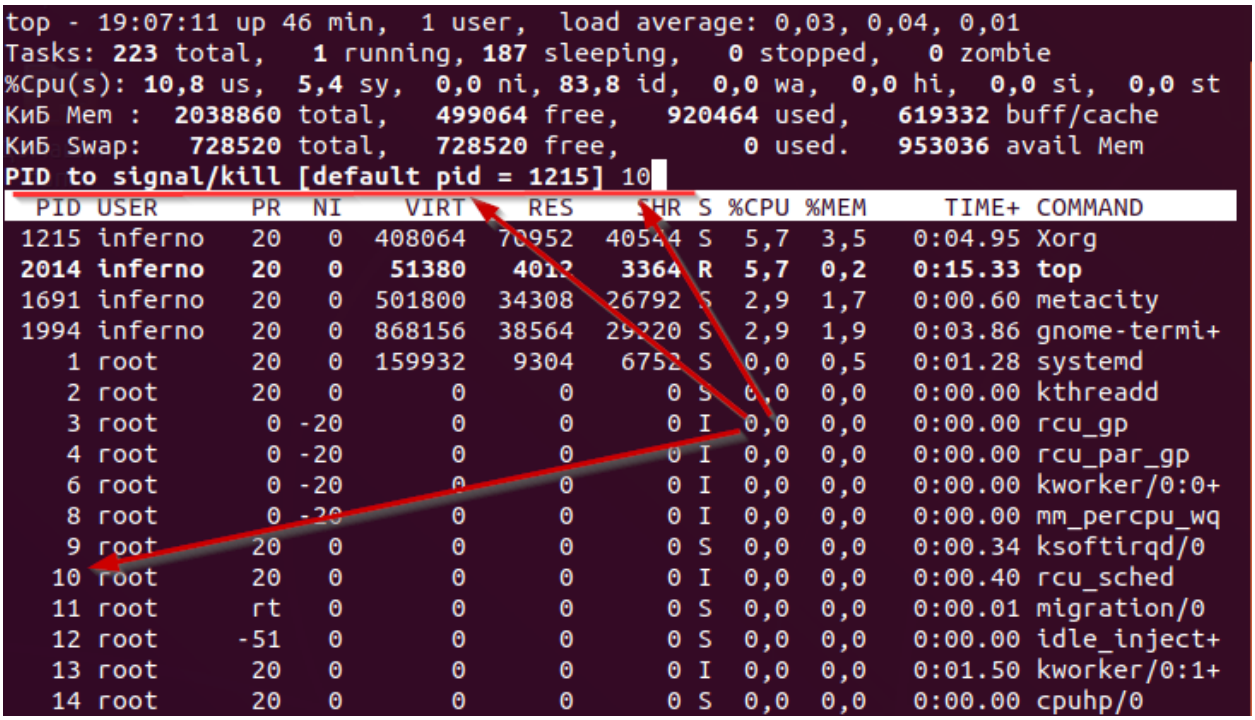

Рисунок 6.21. Снять процесс

Следующая опция попросит указать звуковой сигнал, который оповестит об удачном завершении сервиса. Если первое поле оставить пустым, то утилита закроет самый верхний процесс в списке.

Преимущества Top в том, что он выполняет сортировку в режиме реального времени. Во время его работы нажимаем одну из четырех букв:

- **M** - для сортировки по памяти;
- **P** - для сортировки по использованию CPU;
- **N** - для сортировки по идентификатору процесса;
- **T** - сортировать по времени работы.

Рисунок 6.22. Включение фильтра

Стандартно выборка производится в порядке уменьшения. Для смены отображения – нажмите клавишу R.

6.3.7. Диагностика сетевого подключения (ping, arp, traceroute, dig, nslookup)

Одна из важнейших подсистем, отвечающая за связь любого сервера с внешним миром — сетевая. Через сетевые интерфейсы поступают запросы от удаленных систем и через эти же интерфейсы направляются ответы, что позволяет налаживать коммуникацию и предоставлять/получать сервисы. В связи с этим особенно важно уметь производить диагностику и мониторинг сети хотя бы на базовом уровне, чтобы выявлять проблемы и вносить корректировки в конфигурацию в случае необходимости.

Для операционных систем семейства Linux написано множество утилит, помогающих в диагностике и мониторинге. Познакомимся с наиболее часто используемыми из них.

Согласно стандартам, определяющим работу протокола IPv4, каждое устройство, подключенное к сети, должно иметь как минимум IP-адрес и маску подсети — параметры, которые позволяют уникально идентифицировать устройство в пределах определенной сети. В такой конфигурации устройство может обмениваться сетевыми пакетами с другими устройствами в пределах той же самой логической сети. Если к этому набору параметров добавить адрес шлюза по умолчанию — наш сервер сможет связываться с хостами, находящимися за пределами локального адресного пространства.

В случае каких-либо сетевых проблем в первую очередь проверяем, не сбились ли настройки сетевого интерфейса:

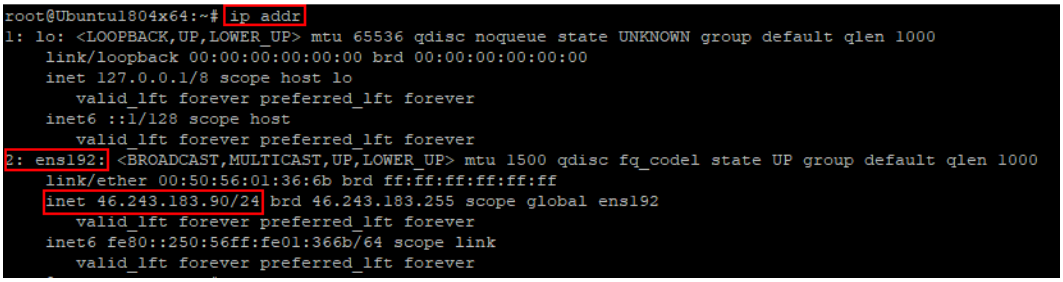

Рисунок 6.23. Проверка настроек сетевого интерфейса

В выводе команды виден перечень сетевых интерфейсов, распознанных операционной системой. Интерфейс lo — это псевдоинтерфейс (loopback). Он не используется в реальных взаимодействиях с удаленными хостами, а вот интерфейс с именем ens192 — то, что нам нужно (именование сетевых интерфейсов различается в разных ветках и версиях ОС Linux). IP-адрес и маска сети, назначенные этому интерфейсу, указаны в поле inet — /24 после адреса обозначают 24-битную маску 255.255.255.0.

Теперь проверим, указан ли шлюз по умолчанию. Команды ip route или route покажут имеющиеся маршруты:

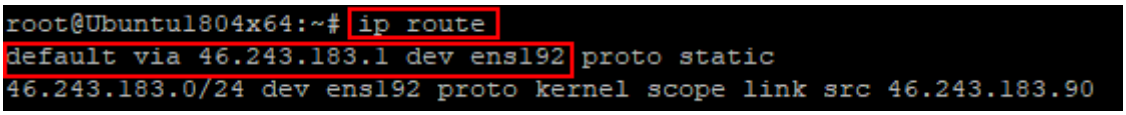

Рисунок 6.24. Проверка маршрута

В таблице маршрутизации мы видим, что имеется маршрут по умолчанию (обозначается либо ключевым словом default, либо адресом 0.0.0.0). Все пакеты, предназначенные для внешних сетей, должны направляться на указанный в маршруте адрес через обозначенный сетевой интерфейс.

Если в настройках интерфейса есть ошибки, их необходимо исправить — помогут в этом другие статьи, для ОС Ubuntu 18.04 или CentOS. Если же все верно — приступаем к диагностике с помощью утилиты ping. Данная команда отправляет специальные сетевые пакеты на удаленный IP-адрес (ICMP Request)

и ожидает ответные пакеты (ICMP Reply). Таким образом можно проверить сетевую связность — маршрутизируются ли сетевые пакеты между IP-адресами отправителя и получателя.

Синтаксис команды ping IP/имя опции:

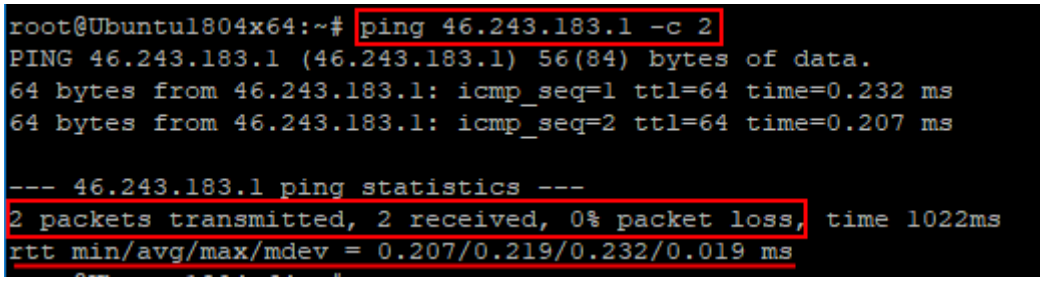

Рисунок 6.25. Синтаксис команды

В данном случае видим, что на оба сетевых пакета, отправленных на адрес нашего шлюза по умолчанию, получены ответы, потерь нет. Это значит, что на уровне локальной сети со связностью все в порядке. Помимо количества полученных/потерянных сетевых пакетов мы можем увидеть время, которое было затрачено на прохождение запроса и ответа – параметр RTT (Round Trip Time). Этот параметр может быть очень важен при диагностике проблем, связанных с нестабильностью связи и скоростью соединения.

Часто используемые параметры:

– `ping -c` количество — указать количество пакетов, которое будет отправлено адресату (по умолчанию пакеты отправляются до тех пор, пока пользователь не прервет выполнение команды. Этот режим можно использовать, чтобы проверить стабильность сетевого соединения. Если параметр RTT будет сильно изменяться в ходе проверки, значит где-то на протяжении маршрута есть проблема);

– `ping -s` количество — указать размер пакета в байтах. По умолчанию проверка производится малыми пакетами. Чтобы проверить работу сетевых устройств с пакетами большего размера, можно использовать этот параметр;

– `ping -I` интерфейс — указать сетевой интерфейс, с которого будет отправлен запрос (актуально при наличии нескольких сетевых интерфейсов и необходимости проверить прохождение пакетов по конкретному сетевому маршруту).

В случае, если при использовании команды ping пакеты от шлюза (или другого хоста, находящегося в одной локальной сети с сервером-отправителем) в ответ не приходят, стоит проверить сетевую связность на уровне Ethernet. Здесь для коммуникации между устройствами используются так называемые MAC-адреса сетевых интерфейсов. За разрешение Ethernet-адресов отвечает протокол ARP (Address Resolution Protocol) и с помощью одноименной утилиты мы можем проверить корректность работы на этом уровне. Запустим команду arp –n и проверим результат:

```
root@Ubuntu1804x64:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
46.243.183.124           ether   00:50:56:01:28:6d   C                     ens192
46.243.183.1             ether   00:50:56:ab:8f:70   C                     ens192
```

Рисунок 6.26. Команда arp-n

Команда выведет список IP-адресов (так как был использован аргумент –n), и соответствующие им MAC-адреса хостов, находящиеся в одной сети с нашим сервером. Если в этом списке есть IP, который мы пытаемся пинговать, и соответствующий ему MAC, значит сеть работает и, возможно, ICMP-пакеты, которые использует команда ping, просто блокируются файрволом (либо со стороны отправителя, либо со стороны получателя).

Часто используемые параметры:

– `arp -n` — вывод содержимого локального arp-кэша в числовом формате. Без этой опции будет предпринята попытка определить символические имена хостов;

– `arp -d адрес` — удаление указанного адреса из кэша. Это может быть полезно для проверки корректности разрешения адреса. Чтобы убедиться, что в настоящий момент времени адрес разрешается корректно, можно удалить его из

кэша и снова запустить ping. Если все работает правильно, адрес снова появится в кэше.

В большинстве случаев в работе с удаленными сервисами мы не используем IP-адреса, а указываем доменные имена удаленных ресурсов. За перевод символических имен в IP-адреса отвечает служба DNS — это сеть серверов, которые содержат актуальную информацию о соответствии имен и IP в пределах доверенных им доменных зон. Самый простой способ проверить работает ли разрешение имен — запустить утилиту ping с указанием доменного имени вместо IP-адреса (например, ping ya.ru). Если ответные пакеты от удаленного сервера приходят, значит все работает как надо. В противном случае нужно проверить прописан ли DNS-сервер в сетевых настройках и удается ли получить от него ответ.

Способы выяснения какой DNS-сервер использует наш сервер различаются в зависимости от используемой версии и дистрибутива ОС Linux. Например, если ОС используется Network Manager для управления сетевыми интерфейсами (CentOS, RedHat и др.), может помочь вывод команды nmcli:

```
[root@CentOS7x64 ~]# nmcli
eth0: connected to eth0
        "VMware VMXNET3"
        ethernet (vmxnet3), 00:50:56:01:37:2E, hw, mtu 1500
        ip4 default
        inet4 46.243.183.101/24
        route4 46.243.183.0/24
        route4 0.0.0.0/0
        inet6 fe80::250:56ff:fe01:372e/64
        route6 ff00::/8
        route6 fe80::/64

lo: unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

DNS configuration:
        servers: 46.243.183.1
        interface: eth0

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(7) manual pages for complete usage details.
```

Рисунок 6.27. Команда nmcli

В настройках сетевого интерфейса, в разделе DNS configuration, мы увидим IP-адрес сервера. В Ubuntu 18.04 и выше, использующих Netplan, используем команду `systemd-resolve —status`.

Используемый сервер также будет указан в настройках интерфейса, в разделе DNS Servers. В более старых версиях Ubuntu потребуется проверить содержимое файлов /etc/resolve.conf и /etc/network/interfaces. Если сервер не указан, воспользуйтесь статьей для ОС Ubuntu 18.04 или CentOS, чтобы скорректировать настройки.

Проверить работу сервиса разрешения имен нам помогут утилиты nslookup или dig. Функционально они почти идентичны: G-вывод утилиты dig содержит больше диагностической информации и гибко регулируется, но это далеко не всегда нужно. Поэтому используйте ту утилиту, которая удобна в конкретной ситуации. Если эти команды недоступны, потребуется доставить пакеты на ***CentOS/RedHat:***

```
yum install bind-utils
```

для ***Debian/Ubuntu:***

```
sudo apt install dnsutils
```

После успешной установки сделаем тестовые запросы:

```
dig ya.ru
```

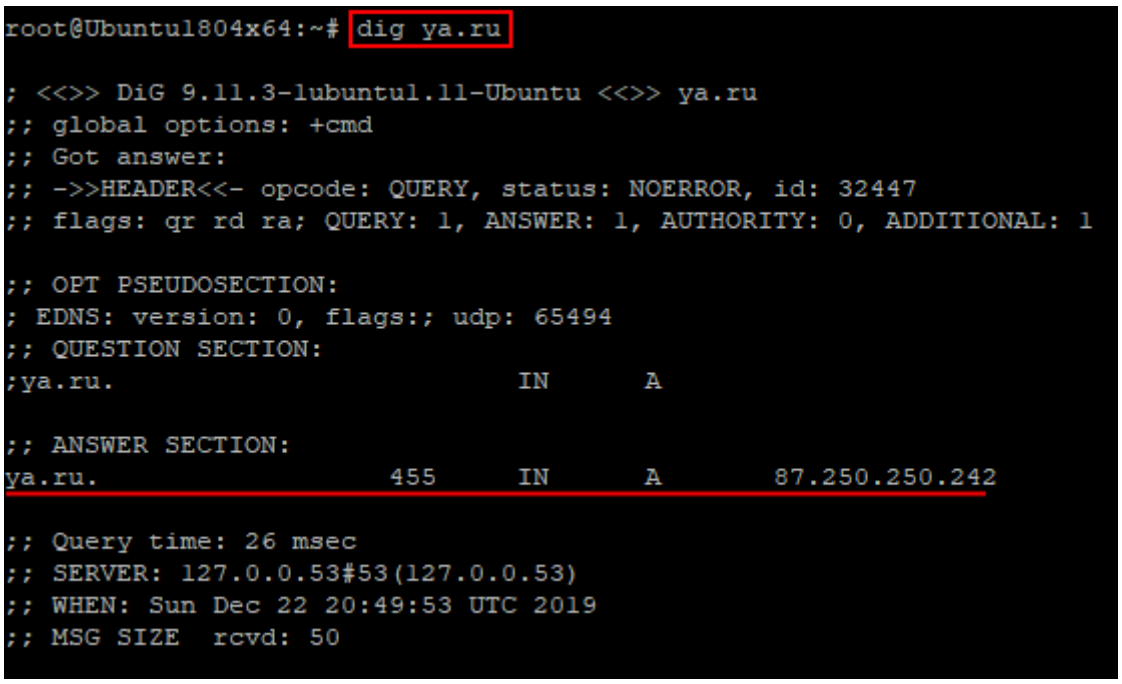

Рисунок 6.28. Тестовые запросы

В разделе Answer Section видим ответ от DNS сервера — IP-адрес для A-записи с доменным именем ya.ru. Разрешение имени работает корректно:

nslookup ya.ru

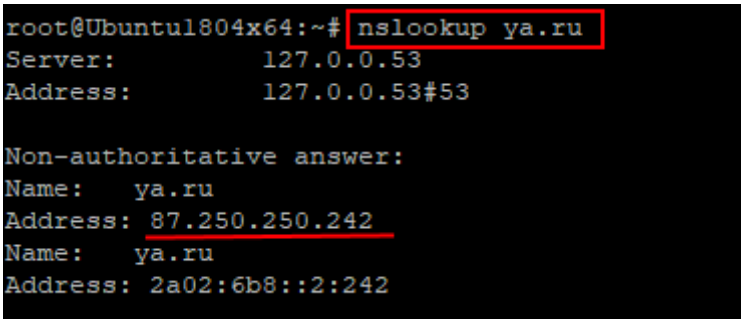

Рисунок 6.29. Подтверждение корректной работы

Аналогичный запрос утилитой nslookup выдает более компактный вывод, но вся нужная сейчас информация в нем присутствует.

Для проверки можно отправить тестовый запрос на другой DNS-сервер. Обе утилиты позволяют эти сделать. Направим тестовый запрос на DNS-сервер Google:

dig @8.8.8.8 ya.ru

nslookup ya.ru 8.8.8.8

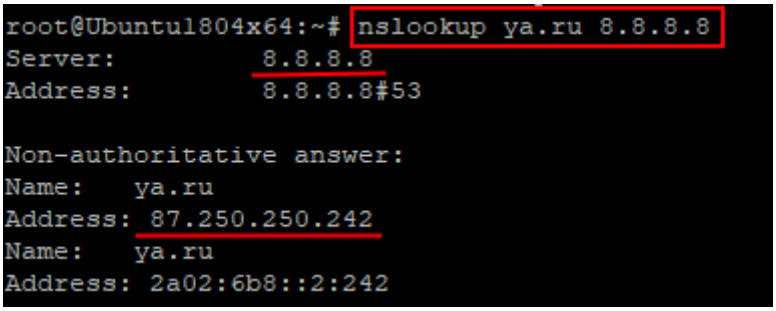

Рисунок 3.20 – Отправка тестового запроса 2

Если имена разрешаются публичным DNS-сервером корректно, а установленным по умолчанию в ОС нет, вероятно, есть проблема в работе этого DNS-сервера. Временным решением данной проблемы может быть использование публичного DNS-сервера в качестве сервера для разрешения имен в операционной системе. В том случае, если разрешение имен не работает ни через локальный, ни через публичный DNS сервер — стоит проверить не блокируют ли правила файрвола отправку на удаленный порт 53 TCP/UDP пакетов (именно на этом порту DNS-серверы принимают запросы).

Часто используемые параметры:

– `nslookup имя сервер` — разрешить доменное имя, используя альтернативый сервер;

– `nslookup -type=тип имя` — получить запись указанного типа для доменного имени (например, nslookup -type=mx ya.ru – получить MX-записи для домена ya.ru);

– `dig @сервер имя` — разрешить доменное имя, используя альтернативый сервер;

– `dig имя тип` — получить запись указанного типа для доменного имени (например, dig ya.ru mx — получить MX-записи для домена ya.ru).

Полный набор опций и параметров для указанных утилит можно найти во встроенной справке операционной системы, используя команду man.

6.3.8. Информация о аппаратном обеспечении Linux

Получение сведений об оборудовании и программном обеспечении вашей Linux-системы является одним из важных этапов администрирования. В этой статье мы углубимся в команды, используемые для поиска параметров системы, применяя инструмент командной строки uname. Кроме того, мы будем использовать команду lshw для получения более подробной информации об оборудовании.

Поскольку команды являются нативными для Linux, они должны работать на любом терминале дистрибутива, использующего ядро Linux.

Для начала выясним детали используемого программного обеспечения, такие как ядро Linux, дистрибутив и другие. Команда, используемая для поиска системной информации — uname. Синтаксис выглядит следующим образом:

```
uname [option]
```

```
[root@CentOS7x64 ~]# uname --help
Usage: uname [OPTION]...
Print certain system information.  With no OPTION, same as -s.

  -a, --all                print all information, in the following order,
                             except omit -p and -i if unknown:
  -s, --kernel-name        print the kernel name
  -n, --nodename           print the network node hostname
  -r, --kernel-release     print the kernel release
  -v, --kernel-version     print the kernel version
  -m, --machine            print the machine hardware name
  -p, --processor          print the processor type or "unknown"
  -i, --hardware-platform  print the hardware platform or "unknown"
  -o, --operating-system   print the operating system
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
For complete documentation, run: info coreutils 'uname invocation'
[root@CentOS7x64 ~]#
```

Рисунок 6.21. Команда uname

Опция `-o` или `—operating-system` отображает используемую операционную систему:

uname -o

```
[root@CentOS7x64 ~]# uname -o
GNU/Linux
```

Рисунок 6.22. Опция operating-system

Опция -n или —nodename отображает имя хоста:

uname -n

Чтобы найти релиз ядра Linux, используйте параметр -r или —kernel-release:

uname -r

Опция -v или —kernel-version выводит версию ядра:

uname -v

Если вам нужна вся вышеуказанная информация в одной команде, используйте параметр -a или —all следующим образом:

uname -a

Рисунок 6.23. Параметр all

Команда List Hardware (lshw) — еще одна популярная команда, используемая системными администраторами для получения очень подробной информации об аппаратных характеристиках машины. Это делается путем чтения различных файлов в каталоге /proc на компьютере с Linux.

Для получения информации утилите lshw необходим root-доступ. Она может сообщить о конфигурации RAM, версии прошивки, конфигурации материнской платы, информацию о процессоре, конфигурации кэша, скорости шины и т.д. Эта команда присутствует по умолчанию в большинстве дистрибутивов Linux. Если приведенные ниже команды не работают, значит утилиты в вашем дистрибутиве нет.

Синтаксис команды:

```
lshw [-format] [-options…]
```

Формат вывода может быть любым из следующих:

– *html* — вывод аппаратной конфигурации в формате HTML;

– *xml* — вывод аппаратной конфигурации в формате XML;

– *json* — вывод аппаратной конфигурации как объект JSON;

– *short* — отображает только основную информацию;

– *businfo* — выводит информацию о шине.

Параметры могут быть любыми из следующих:

– `-class CLASS` — показывает только определенный класс оборудования;

– `-C CLASS` — такой же, как ‘-class CLASS’;

– `-c CLASS` — такой же, как ‘-class CLASS’;

– `-disable TEST` — отключить тест (например, pci, isapnp, cpuid и т.д.);

– `-enable TEST` — включить тест (например, pci, isapnp, cpuid и т.д.);

– `-quiet` — не отображать статус;

– `-sanitize` — удалить конфиденциальную информацию, такую как серийные номера и т.д.;

– `-numeric` — числовые идентификаторы (для PCI, USB и т.д.);

– `-notime` — исключить изменяемые атрибуты (временные метки) из вывода.

Поскольку команда должна выполняться с правами root, используйте sudo перед командой:

```
sudo lshw -short
```

7. АДМИНИСТРИРОВАНИЕ ЭЛЕМЕНТОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИ ИСПОЛЬЗОВАНИИ ТЕХНОЛОГИЙ ВИРТУАЛИЗАЦИИ

7.1. Общие положения администрирования виртуальной инфраструктуры

В большинстве случаев виртуальная инфраструктура ведет себя точно также, как и физическая, но технологии виртуализации имеют свои особенности. Шаблоны виртуальных машин позволяют создавать полностью настроенный виртуальный сервер с установленной операционной системой за несколько кликов мыши из панели управления облачного провайдера.

7.1.1. Миграция физических серверов в виртуальную среду

Обычно к моменту принятия решения о разворачивании виртуальной инфраструктуры организация уже имеет физическую инфраструктуру, включающую физические серверы и телекоммуникационную аппаратуру. Физические серверы можно превратить в виртуальные, при помощи различных специализированных инструментов:

- команда dd в Linux;
- программа disk2vhd в Windows;
- программа VMware Converter Standalone;
- специализированные коммерческие продукты.

Есть системы, которые не стоит мигрировать в облака:

1. Программы, для работы которого требуется физическое оборудование.

2. Такое оборудование, как ключи электронной защиты, используемое для защиты от несанкционированного копирования, невозможно виртуализовать. Иногда такие проблемы можно решить путем перехода на электронный ключ.

3. Системы, для которых нужна высокая производительность.

4. Приложения и устройства, которые загружают оперативную память, жесткий диск, процессор, видеокарта, не подходят для миграции.

5. ПО с лицензионными ограничениями на виртуализацию.

6. Некоторые лицензионные соглашения запрещают использовать продукты в виртуальной среде.

7. Критически важные приложения для без тестирования.

8. Не стоит переводить жизненно важные для организации системы или приложения на виртуальную платформу без предварительного тестирования.

9. ИТ-системы, требующие повышенной безопасности.

10. Виртуализация любой системы, содержащей конфиденциальные данные, может предоставлять угрозу ИТ-безопасности.

При использовании арендованной виртуальной инфраструктуры (IaaS) вам не придется следить за обновлением программного обеспечения серверов, и коммуникационного оборудования. Но самим виртуальным машинам требуются регулярные обновления и операционных систем, и приложений. Обычной проблемой в крупной инфраструктуре является размножение образов. Это размножение приводит к тому, что появляется огромное количество шаблонов виртуальных машин с одной и той же операционной системой, но с различными приложениями или ролями. Управлять таким зоопарком образов и обеспечивать их обновление непросто, но такие продукты, как System Center Virtual Machine Manager позволяют сократить число образов и устанавливать приложения, роли и функции в виртуальных серверах при помощи пакетов приложений.

7.1.2. Установка SSL-сертификатов

Если ваша организация разрабатывает или использует веб-приложения, расположенные в виртуальной инфраструктуре, то доступ к этим приложениям необходимо защищать при помощи шифрования. Можно, конечно, использовать самоподписанные сертификаты или бесплатные сертификаты Let’s Encrypt, но у них есть существенные недостатки:

1. Ограниченный срок действия.

2. Бесплатные сертификаты Lets’ Encrypt рассчитаны на 90 дней. Для автоматического перевыпуска необходимо настроить планировщик, который будет запускать скрипт автоматического перевыпуска или принимать настройки центра сертификации, которые смогут вносить изменения в конфигурацию ваших серверов.

3. Не все домены можно защищать сертификатами Let’s Encrypt.

4. Эти сертификаты.

5. Отсутствие технической поддержки.

6. На сайте Let’s Encrypt есть только документация на английском языке. Некоторые вопросы выпуска и использования есть на тематических форумах.

7. Нет расширенных проверок.

8. При выпуске сертификата проверяется только принадлежность домена клиента. Таким образом, кто угодно в интернете может создать сертификат с названием любой организации.

9. Проблемы с поддержкой в браузерах.

10. При использовании сертификатов Let’s Encrypt некоторые браузеры выводят сообщения о том, что браузер не доверяет этому сертификату. В основном, это происходит при использовании устаревших браузеров и операционных систем.

11. Самоподписанные сертификаты выводят сообщение «Сертификат безопасности не является доверенным».

12. Эта надпись будет отпугивать потенциальных клиентов. Такие сертификаты можно испольльзовать только при обмене данными между пользователями, которые знают о самоподписанном SSL-сертификате и подтвердили его в браузере.

13. Самоподписанные сертификаты не подтверждают реально компании.

14. Они лишь обеспечивают безопасную передачу данных.

Доверенные сертификаты – единственное надежное решение для защиты веб-приложений. Провайдеры облачных услуг могут предложить выпуск доверенных сертификатов SSL из панели управления.

7.1.3. Политики безопасности

Виртуальная инфраструктура повышает степень интеграции ИТ-средств, при этом одновременно уменьшая количество физического оборудования. Но количество сетевых приложений, сервисов остается таким же и даже большим. Поэтому защищать виртуальную инфраструктуру нужно комплексно, комбинируя сетевые и локальные средства защиты. Можно использовать следующие защитные механизмы:

– средства сетевой аутентификации и авторизации пользователей;

– межсетевое экранирование как внутри сервера виртуализации между гостевыми машинами, так и по периметру виртуальной инфраструктуры;

– системы регистрации, сбора и корреляционного анализа событий безопасности;

– средства разграничения доступа и делегирование полномочий к виртуальным машинам

– системы контроля целостности конфигураций распределенных компонентов виртуальной инфраструктуры;

– средства антивирусной защиты;

– средства управления доступом к элементам виртуальной инфраструктуры.

Следует иметь ввиду, что политика безопасности, привязанная к таким физическим атрибутам, как физический сервер, IP-адрес, MAC-адрес не работает в облаке, так как провайдер облачных услуг может их изменить. Политику безопасности следует привязать к логическим атрибутам. Становятся важными идентификация, групп или роль пользователей, а также чувствительность нагрузки.

Модель контроля доступа должна быть основана на пользователе. Контроль сетевого доступа (Network Access Control — NAC) определяет пользователей и к чему у пользователей есть доступ. Контроль доступа должен учитывать, что пользователи могут прийти из любого места в Интернет, в любое время, с любого устройства, а также запрашивать доступ одновременно с разных устройств.

7.1.4. Антивирусная защита

Технологии виртуализации порождают ряд новых рисков информационной безопасности. Типичным примером атаки на облачные среды является «антивирусный шторм», который вызывает одновременный запуск множества антивирусов. Такой шторм может привести к нарушению работоспособности виртуальной машины. Высокая доступность виртуальных машин и приложений может создать множество возможностей для злоумышленников. При этом одна незащищённая или зараженная виртуальная машина может стать источником угрозы для всей виртуальной инфраструктуры.

Традиционные средства защиты, например, агентские антивирусы, не всегда применимы в условиях виртуализации. Стоит выделить три основных подхода к антивирусной защите виртуальной инфраструктуре.

Первый подход состоит в том, что виртуальная среда предоставляет специальный программный интерфейс для контроля виртуальных машин через гипервизор, а антивирус используется им, перенаправляя всю защиту на специализированную виртуальную машину. Это позволяет отказаться от использования антивирусных агентов на виртуальных машинах.

Второй подход заключается в работе по старой схеме с использованием антивирусных агентов, которые нужно обновлять и настраивать. Но вместе с тем производители антивирусов стараются предоставить новые возможности для оптимизации исполнения агентов в виртуальной среде.

Третий подход заключается в том, чтобы не отказываться от агентов полностью, делать их максимально легковесными и простыми для исполнения, но в то же время большую часть аналитики реализовывать на специализированной виртуальной машине. Этот более универсален, но в некоторых случаях может уступать первым и вторым подходам.

7.1.5. Системный мониторинг и мониторинг производительности

Так как виртуальная машина работает с виртуальным аппаратным обеспечением, за которым нет необходимости следить также, как и за физическим, можно удалить все программы-агенты аппаратного обеспечения при переводе физического сервера на виртуальную платформу. Кроме того, виртуальные машины загружаются быстрее физических, и из-за этого система мониторинга может не успеть обнаружить перезагрузку виртуальной машины, если ее интервал сбора данных окажется слишком большой.

Можно выделить следующие критерии работоспособности, которые будут собирать системы мониторинга:

Традиционные системы сбора данных о производительности часто некорректно работают на виртуальных машинах, поэтому для мониторинга необходимо использовать специализированные средства, которые могут быть как встроенными в платформу виртуализации, так и разработанными специально для мониторинга виртуальной инфраструктуры.

Для оценки производительности виртуальных машин используются следующие критерии:

Для виртуальных машин Windows:

- доступный объем памяти, мегабайт;
- среднее время на операцию записи, секунд (логический диск);
- среднее время на операцию записи, секунд (диск);
- среднее время на операцию чтения, секунд (логический диск);
- среднее время на операцию перемещения, секунд (логический диск);

– среднее время на операцию чтения, секунд (диск);
– среднее время на операцию перемещения, секунд (диск);
– текущая длина очереди диска (логический диск);
– текущая длина очереди диска (диск);
– процент времени простоя диска;
– ошибка или повреждение файловой системы;
– мало свободного пространства на логическом диске (в процентах);
– мало свободного пространства на логическом диске (в мегабайтах);
– процент времени простоя логического диска;
– страниц памяти в секунду;
– процент используемой пропускной способности чтения;
– процент используемой общей пропускной способности;
– процент используемой пропускной способности записи;
– процент используемой выделенной памяти;
– процент времени простоя диска;
– работоспособность службы DHCP-клиента;
– работоспособность службы DNS-клиента;
– работоспособность службы RPC;
– работоспособность службы сервера;
– общий процент использования ЦП;
– работоспособность службы журнала событий Windows;
– работоспособность службы брандмауэра Windows;
– работоспособность службы удаленного управления Windows.

Для виртуальных машин Linux:

– время обращения к диску (с)
– время чтения с диска (с)
– время записи на диск (с)
– работоспособность диска;
– свободное пространство на логическом диске;

– свободное пространство на логическом диске (в процентах);
– свободные дескрипторы на логическом диске (в процентах);
– работоспособность сетевого адаптера;
– общее процессорное время в процентах;
– доступный для операционной системы объем памяти, мегабайт.

7.1.6. Резервное копирование

Виртуализация усложняет задачу резервного копирования. С одной стороны, можно продолжать выполнять резервное копирование с помощью традиционных средств путем установки на каждую виртуальную машину агента, который будет копировать файлы в нужное место. Этот метода надежен, но в условиях виртуальной среды может работать некорректно. При одновременном запуске агентов на всех виртуальных машинах возникает «шторм», в ходе которого каждая виртуальная машина стремится захватить все ресурсы сервера для выполнения задачи резервного копирования.

Другой подход состоит в том, что резервное копирование может выполняться на уровне сервера виртуализации. В этом случае копирование данных из виртуальных машин может проходить гораздо быстрее.

Одной из основных задач системного администратора является резервное копирование данных, находящихся на рабочих станциях и ноутбуках. Ведь в случае хищения ноутбука данные могут попасть к злоумышленникам, а пользователи редко сами принимают шифрование.

Особого внимания заслуживает тема дедупликации. Как мы уже писали, новые виртуальные машины часто создаются из шаблонов, в результате чего данные на дисках виртуальных машин в значительной степени совпадают. Дедупликация позволяет находить одинаковые блоки на дисках и в резервной копии сохранять только один экземпляр.

Если используется технология VDI (инфраструктура рабочих столов), все обрабатываемые данные будут находиться в виртуальных серверах или

хранилищах на стороне облачного провайдера. Таким образом, резервное копирование производится в облаке и в случае кражи клиентского устройства данные не попадут в чужие руки.

7.1.7. Оптимизация виртуальных дисков, перенос виртуальных серверов и настройка многофакторной аутентификации

Подобно тому, как на физических серверах необходимо производить дефрагментацию жестких дисков, в виртуальных машинах необходимо производить сжатие виртуальных дисков. Сжатие (compact) применяется к динамическим расширяемым и разностным виртуальным жестким диска. Команда Optimize-VHD уменьшает размер VHD-файла, удаляя пустое пространство, оставшееся после удаления данных с виртуального жесткого диска. Кроме того, эта команда перестраивает блоки для более эффективного использования дискового пространства, что также уменьшает размер VHD файлов.

Что делать, если уже есть виртуальная инфраструктура в облаке, но по каким-то причинам ее нужно перенести к другому провайдеру? Здесь поможет функция экспорта виртуальных машин, . При переносе виртуальных машин из облака Amazon S3 можно воспользоваться корзиной (bucket) в панели управления Amazon S3.

Если виртуальная инфраструктура была создана в облаке Microsoft Azure, придется использовать специальный командлет PowerShell Save-AzureVhd, а если в Hyper-V, то встроенной функцией экспорта оснастки «Управление Hyper-V».

Также можно экспортировать виртуальный сервер в файл формата OVA/OVF(открытый стандарт для хранения и распространения виртуальных машин), а потом импортировать полученный файл с помощью панели управления нового провайдера.

Для безопасного доступа к облачной инфраструктуре одного пароля уже недостаточно. Многофакторная проверка подлинности предоставляет дополнительную проверку подлинности в дополнение к учетным данным пользователя. К возможным опциям многофакторной подлинности относятся:

- мобильные приложения;
- телефонные звонки;
- текстовые сообщения.

Несмотря на то, что администратору виртуальной инфраструктуры больше не нужно заботиться о физическом оборудовании, забот у него не становится меньше. Администрирование виртуальной инфраструктуры может потребовать большего внимания, чем физической, поскольку инфраструктура усложняется, добавляется множество задач. Виртуализация несет в себе как новые возможности, таки и новые риски, которые необходимо учитывать.

7.2. Инструменты виртуализации в области администрирования подсистем обеспечения информационной безопасности

7.2.1. Серверная виртуализация и ее функции

Смысл серверной виртуализации заключается в запуске нескольких (а не одной) гостевых ОС на одном сервере одновременно. Цель подобного запуска — из одного физического сервера сделать несколько виртуальных. Каждая из таких гостевых ОС имеет свои индивидуальные ресурсы. За них отвечает платформа виртуализации с физического хоста.

Рисунок 7.1. Пример серверной виртуализации

Процесс виртуализации дает возможность сокращать численность физических серверов. То есть благодаря такому процессу устанавливается не несколько старых серверов, а один мощный. При этом необходимое количество гостевых ОС, которые обеспечены логической изоляцией, запускается в виртуальной среде. Она предполагает гибкость в администрировании инфраструктуры с высокими параметрами отказоустойчивости. Отказоустойчивость представляет собой «переезд» виртуальных машин с одного сервера на другой, в случае выхода из строя одного из них. Технологии виртуализации помогают произвести описанную процедуру, не останавливая работу всего сервиса, так что пользователи этого не заметят.

Корпоративные клиенты находят для себя в технологиях виртуализации, применяемых к серверам, следующие преимущества:

– сокращение финансовых затрат на построение ЦОДов, закупку оборудования – как серверного, так и сетевого;

– упрощение процесса администрирования;

– исключение простоя оборудования.

Еще один из видов технологии виртуализации, набирающий все большую популярность, — это виртуализация приложений, которая предполагает изоляцию отдельно взятого приложения от ресурсов ОС.

Для многих организаций привлекательной является контейнеризация — упаковка отдельных приложений в виртуальный образ. В этом виде его легко можно развернуть в облачной виртуальной среде и эластично масштабировать.

7.2.2. Контейнерная виртуализация

Специалисты выделяют два технологических подхода к виртуализации программного обеспечения серверов:

– гипервизорная виртуализация (при использовании полноценных виртуальных машин, с виртуализацией на уровне оборудования);

– контейнерная виртуализация (с плотным размещением виртуальных сред на сервере, с более быстрыми откликами системы на запросы пользователей).

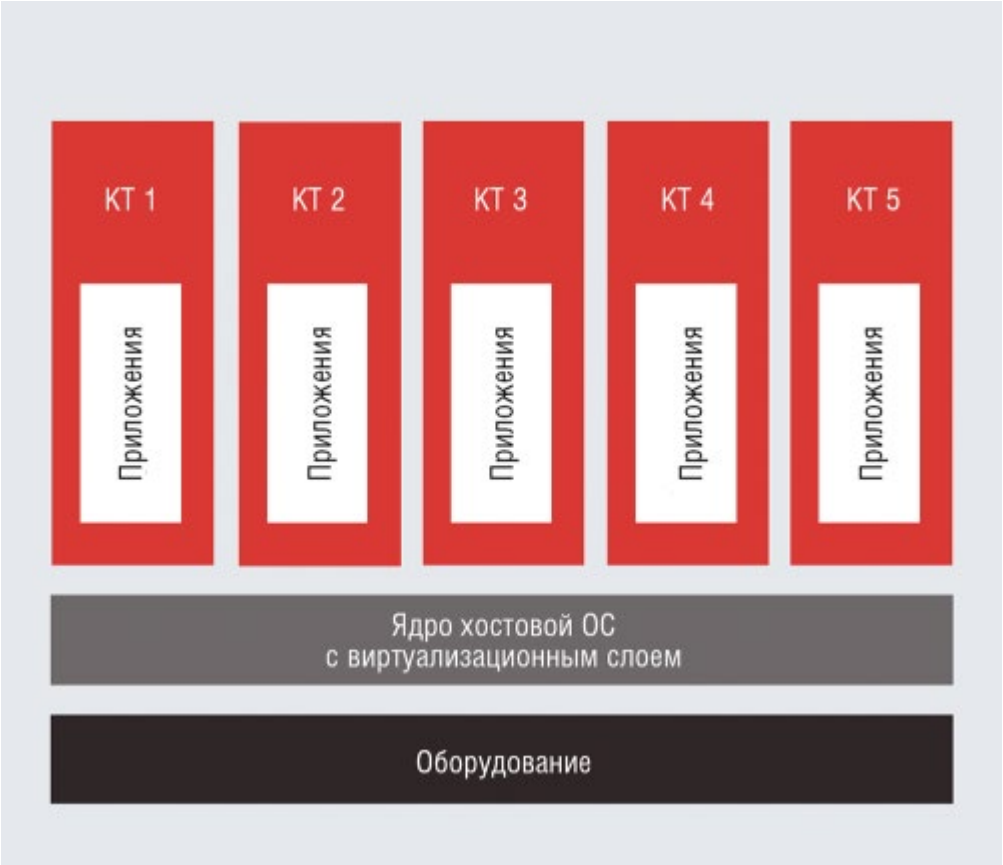

Рисунок 7.2. Пример контейнерной виртуализации

Особенности контейнерной виртуализации заключаются в следующем:

– «гости» пользуются одним ядром «хозяина»;

– Web-сервисы, упакованные в контейнеры, обслуживают большее количество запросов клиентов;

– при этом нет необходимости в подключении дополнительного оборудования.

К «плюсам» контейнерной виртуализации специалисты относят высокую производительность Web-сервисов – на 50% больше, чем при использовании гипервизорных систем.

Сейчас существуют платформы комбинированного типа – на базе контейнерных и гипервизорных систем. Преимущество подобной комбинации очевидно — можно гибко менять соотношение между этими системами в ЦОДе, при этом осуществлять управление ними, используя единый инструмент.

Рисунок 7.3. Пример платформы комбинированного типа

7.2.3. Пример организации виртуального хранилища средствами Docker-контейнеров

Виртуализация на основе процессов Docker имеет много преимуществ, особенно в сочетании с преимуществами слоев образов. Это позволяет невероятно быстро производить генерацию контейнеров и использовать легкий вес ресурсов. Однако одним из побочных эффектов эфемерной модели процесса Docker является то, что вы должны обязательно планировать, когда и как вы хотите сохранить постоянные данные.

Предположим, что мы заходим в оболочку внутри busybox контейнера:

```
docker run -it --rm busybox
```

Затем, давайте запишем некоторые данные, скажем, в /tmp:

```
echo "Data!" > /tmp/data
cat /tmp/data
Data!
```

Мы видим, что данные определенно записываются. Но куда же на самом деле идут эти данные? Как мы узнали ранее, образы Docker состоят из слоев, уложенных друг на друга, чтобы привести к окончательному образу. Каждый из этих слоев содержит данные, измененные в такой операции, как установка инструмента, добавление исходного кода и т.д. Каждый из этих слоев становится доступным только для чтения после его создания. Когда контейнер создается из образа, тонкий R/W слой добавляется поверх предыдущих слоев образа. Этот слой обрабатывает все вызовы записи из контейнера, которые, в противном случае, были бы направлены на слои ниже, доступные только для чтения. Помните, что контейнеры эфемерны по своей природе. Они предназначены для того, чтобы иметь определенную продолжительность жизни и умереть в какой-то момент, как и любой процесс. Тонкий слой чтения/записи также эфемерен — он исчезает вместе с контейнером. Таким образом, любые записи, которые мы выполняем в контейнере, ограничены временем жизни этого контейнера. Они исчезнут, когда контейнер будет уничтожен. Это очевидное ограничение, которое не способствует хранению статусной информации. Итак, как разработчики и администраторы работают с этим? Они используют Тома Docker.

Тома Docker — это способ создания постоянного хранилища для контейнеров Docker. Тома Docker не привязаны к времени жизни контейнера, поэтому сделанные в них записи не исчезнут, как это произойдет с контейнером. Они также могут быть повторно подключены к одному или к нескольким контейнерам, чтобы можно было обмениваться данными и подключать новые контейнеры к существующему хранилищу. Тома Docker работают путем создания каталога на главной машине и последующего монтирования этого каталога в контейнер (или в несколько контейнеров). Этот каталог существует вне многослойного образа, который обычно содержит контейнер Docker, поэтому он не подчиняется тем же правилам (только для чтения и т. д.).

Давайте создадим том Docker и посмотрим его в действии:

```
docker volume create
1d358c3fc3750f98345713eee5c294dee52...
```

Простой вызов docker volume create позволит создать новый том. Если мы проверим этот том, мы можем увидеть, где он находится на хост-файловой системе:

```
docker                          volume                                inspect
1d358c3fc3750f98345713eee5c294dee526a3f5d0bd41a0ff4d117218c4af73
[
{
"CreatedAt": "2020-05-17T16:25:30Z",
"Driver": "local",
"Labels": {},
"Mountpoint":
"/var/lib/docker/volumes/1d358c3fc3750f98345713eee5c294dee526a3f5d0
bd41a0ff4d117218c4af73/_data",
"Name":
"1d358c3fc3750f98345713eee5c294dee526a3f5d0bd41a0ff4d117218c4af73",
"Options": {},
"Scope": "local"
}
]
```

Листинг 7.1. Информация о Docker

С вызовом inspect приходит много информации, но все, что нас действительно беспокоит прямо сейчас, это Mountpoint. Обратите внимание, что здесь указан путь, начианющийся с /var/lib/docker…. Если вы откроете этот путь на своем компьютере, на котором работает Docker, вы можете просмотреть данные, хранящиеся внутри этого тома. Метод, который мы только что использовали для создания тома, не является единственным способом. При запуске контейнера вы можете указать -v, чтобы создать новый том на лету:

```
docker run -it --rm -v testdata:/data busybox
```

Как вы можете видеть, мы добавили новый аргумент к нашей docker run команде: -v. Существует специальный синтаксис для этого аргумента, с полями, разделенными двоеточиями. Первое поле — это название тома, в данном случае, testdata. Второе поле — это путь в контейнере, куда том будет примонтирован, в нашем случае, /data. Давайте запишем данные на том из контейнера:

echo "Hello!" > /data/hello

Эти данные видны снаружи контейнера, в пути монтирования тома на хосте:

pwd

cat hello

/var/lib/docker/volumes/testdata/_data

Hello!

Вы должны были заметить, что в пути название тома больше не является случайной строкой — это теперь имя тома, которое мы указали при использовании -v аргумента. Тома Docker могут иметь либо рандомизированные имена, инициализированные Docker Engine, или имена, назначенные пользователем самостоятельно. Имена должны быть уникальными для каждого хоста. Созданный во время выполнения том Docker теперь становится доступным в команде docker volume ls:

```
docker volume ls
DRIVER VOLUME NAME
local 1d358c3fc3750...
local c63a0af5c8282...
local testdata
```

Листинг 7.2. Вывод docker volume

Это означает, что мы можем использовать этот том снова с другим контейнером или даже несколькими контейнерами. Давайте проверим это прямо сейчас. Во-первых, подключите том к busybox контейнеру:

```
docker run -it --rm -v testdata:/data busybox
```

Внутри контейнера давайте напечатаем информацию о системе, а затем запишем на том:

```
uname -a
echo "Hello 2" > /data/hello
Linux 7e299450b997 4.9.125-linuxkit #1 SMP Fri Sep 7
08:20:28 UTC 2018 x86_64 GNU/Linux
```

Теперь запустите второй busybox контейнер, работающий одновременно с первым:

```
docker run -it --rm -v testdata:/data busybox
```

Нам доступны данные, которые были записаны в первом контейнере:

```
uname -a
```

```
    cat /data/hello
    Linux c5bf9ca04d3a 4.9.125-linuxkit #1 SMP Fri Sep 7
08:20:28 UTC 2018 x86_64 GNU/Linux
    Hello 2
```

Это подчеркивает еще одно из преимуществ томов Docker: совместное использование данных между контейнерами.

Есть много допустимых вариантов использования для контейнеров Docker, но здесь мы рассмотрим два наиболее распространенных. Для каждого из этих примеров давайте представим, что у нас есть простое приложение, которое работает и собирает данные с некоторых датчиков погоды. Мы хотим собрать кучу метеорологических показателей, сохранить их, а затем использовать их снова в будущем. Давайте назовем наше тестовое приложение WeatherMon. Если мы выполняем WeatherMon без использования томов Docker, все данные, которые мы собираем, будут уничтожены при исчезновении контейнера. Нам это не сильно поможет, если наша цель — собрать данные и сделать их доступными для дальнейшего использования. Тома Docker удобны тем, что мы можем сохранить наши данные в томе и оставить их за пределами срока службы контейнера. Допустим, мы создаем наш контейнер, вызывая docker run с аргументом -v weathermon:/opt/weathermon. Тогда наше приложение может хранить свои метрики погоды в каталоге /opt/weathermon непосредственно на эфемерном слое чтения/записи, предоставленном контейнером. Мы также могли бы настроить удаленную базу данных для хранения этой информации, но тома Docker предоставляют альтернативу для хранения локально используемых данных.

Предположим, что мы уже используем контейнер WeatherMon некоторое время и собрали довольно много данных. Мы хотим проанализировать эти данные, чтобы определить информацию, такую как средняя температура за день, или какая неделя в месяце имела самую высокую среднюю влажность. Используя тома Docker, мы можем смонтировать существующий том в новый контейнер, WeatherMon-Analytics. Этот новый контейнер может считывать данные, не

прерывая их сбор контейнером WeatherMon. Затем он может выполнять аналитику, которую мы хотим, и хранить эту информацию в том же томе или в другом томе, если это необходимо.

Есть два других типа томов Docker: bind mount и tempfs mount.

Bind mount используются для монтирования существующего пути на хосте в контейнер. Используя —mount совместно с <host path>:<container path>, вы можете указать существующие каталоги, которые будут монтироваться в контейнер. Это очень удобно при использовании информации о конфигурации, такой как каталоги внутри /etc. Это также полезно, когда у вас есть информация, которую вы хотите использовать в контейнере, например, существующие наборы данных или статические файлы веб-сайта.

Задача tempfs монтирования состоит в том, чтобы обеспечить доступное для записи расположение, которое специально не сохраняет информацию после окончания срока службы контейнера. Возможно, вы думаете: «зачем это нужно?” В контейнере, который не имеет подключенного тома, все записи идут в тонкий слой R/W, вставленный во время выполнения. Любая запись, направленная на этот слой, влияет на файловую систему, поскольку эти записи выполняются на базовом хосте. Обычно это не является проблемой, если вы не пишете значительные объемы одноразовых данных (таких как журналы). В этом случае вы можете наблюдать снижение производительности, так как файловая система должна обрабатывать все эти вызовы write. tempfs монтирование было создано для предоставления контейнерам временного пути записи, который не влияет на операции файловой системы. В частности, tempfs это эфемерное монтирование, которое записывается непосредственно в память. Этот том можно создать с помощью —tempfs аргумента.

По умолчанию тома хранят информацию о базовой хост-системе. Docker также имеет концепцию, называемую драйверами томов, которая позволяет указать, как и где хранить тома. Например, вы можете хранить том Docker внутри корзины Amazon S3. Это может быть удобно, если вы хотите, чтобы информация

сохранялась не только за пределами срока службы контейнера, но и за пределами срока службы хоста.

7.2.4. Миграция виртуальной машины VMware

Для миграции вам необходимо сконвертировать ваш виртуальный сервер в файл формата ovf или ova.

OVF (Open Virtualization Format) — стандарт для хранения и распространения виртуальных машин. OVA является TAR архивом каталога с пакетом OVF.

Требования к ova/ovf файлам:

1. Тип дисков scsi.

2. Hardware version не выше 13.

Если вы используете один из продуктов компании VMware для виртуализации ваших серверов и/или настольных компьютеров, например VMware Workstation, VMware Fusion, VMware Server, VMware Player, процесс конвертации не будет трудоемким.

Все продукты поддерживают стандартный экспорт машин в необходимый формат.

7.2.4.1. VMware Workstation

Необходимые условия:

1. Убедитесь в том, что виртуальная машина не зашифрована. Вы не можете экспортировать зашифрованную виртуальную машину в формат OVF.

2. Убедитесь в том, что виртуальная машина выключена.

Процедура:

1. Выберете виртуальную машину и далее File > Export to OVF.

2. Введите имя для ovf-файла и выберите директорию для сохранения файла.

3. Нажмите Save для начала процесса экспорта в OVF.

4. Процесс может занять несколько минут. Строка состояния показывает ход процесса экспорта.

7.2.4.2. VMware Fusion

Необходимые условия:

1. Убедитесь в том, что виртуальная машина не зашифрована. Вы не можете экспортировать зашифрованную виртуальную машину в формат OVF.

2. Убедитесь в том, что виртуальная машина выключена.

Процедура:

1. Выберете виртуальную машину и далее File > Export to OVF.

2. Введите имя для ovf-файла и выберите директорию для сохранения файла.

3. Нажмите Save для начала процесса экспорта в OVF.

4. Процесс может занять несколько минут. Строка состояния показывает ход процесса экспорта.

7.2.4.3. vSphere Client

1. Выберете виртуальную машину и далее File > Export > Export OVF Template.

2. В диалоговом окне “Export OVF Template” введите имя шаблона. Примечание: при экспорте шаблона OVF с именем, которое содержит звездочки (*), эти символы превращаются в символы подчеркивания (_).

3. Выберите директорию для сохранения файла шаблона, или кликните “…” для выбора местоположения.

4. В поле Format выберите тип OVF или OVA. OVF — выберете данный формат, чтобы сохранить шаблон в виде набора файлов (.ovf, .vmdk и .mf) Этот формат является оптимальным, если вы планируете опубликовать файлы на веб-

сервере или в хранилище дисков. Пакет можно импортировать, например, в клиенте vSphere путем публикации URL в .ovf файл. OVA — выберете данный формат для более удобного распространения пакета OVF в виде отдельного файла, если его нужно явно загрузить с веб-сайта или перемещены с помощью USB. Фактически является архивом формата OVF.

5. В поле Description введите описание виртуальной машины. По умолчанию текст из панели Notes со вкладки Обзор виртуальной машины появится в этом текстовом поле.

6. Установите флажок, если вы хотите включить графические файлы, прикрепленные к floppy и CD / DVD-устройств в пакет OVF. Примечание: этот флажок виден только если к виртуальной машине подключен файл ISO или к дисководу подключен образ.

7. Строка состояния показывает ход процесса экспорта.

7.2.4.4. VMware OVF Tool

VMware OVF Tool представляет собой решение для импорта и экспорта OVF пакетов из продуктов VMware. Мы будем использовать данную утилиту для конвертации файла vmx в файл формата ovf, т.к. она позволяет создавать виртуальные модули из виртуальных машин.

Загрузите установщик VMware OVF Tool на сайте компании Vmware. Выберете необходимую версию.

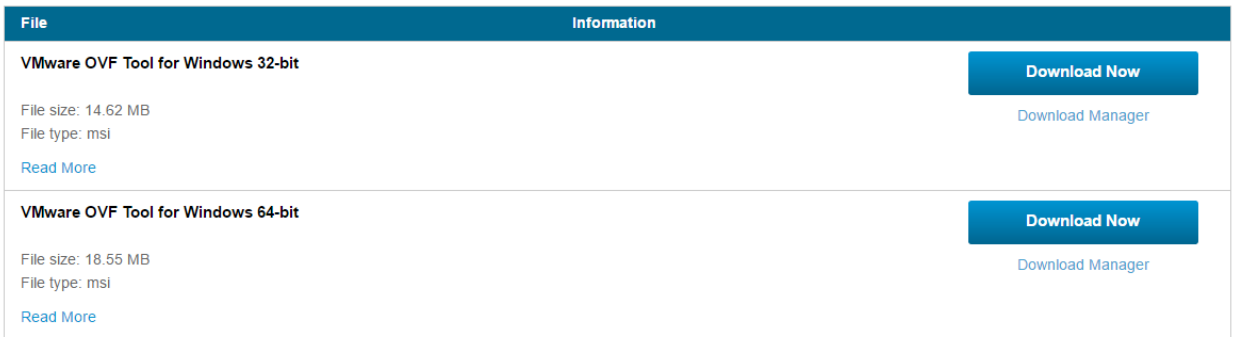

Рисунок 7.4. Выбор версии VMware OVF Tool

Перейдите в папку «Downloads» и запустите на исполнение скачанный файл. После установки откройте Windows PowerShell. Для конвертации необходим файл ovftool.exe, который находится по указанному при установке пути. Например:

```
C:Program FilesVMwareVMware OVF Toolovftool.exe
```

Перейдите в каталог «VMware OVF Tool» с помощью команды cd:

```
cd 'C:Program FilesVMwareVMware OVF Tool'
```

В командной строке введите команду со следующим синтаксисом:

```
.ovftool.exe <путь до файла vmx> <путь до файла ovf>
```

Например:

```
.ovftool.exe "C:VMVM-exampleVM-example.vmx" "C:VMVM-exampleVM-example.ovf"
```

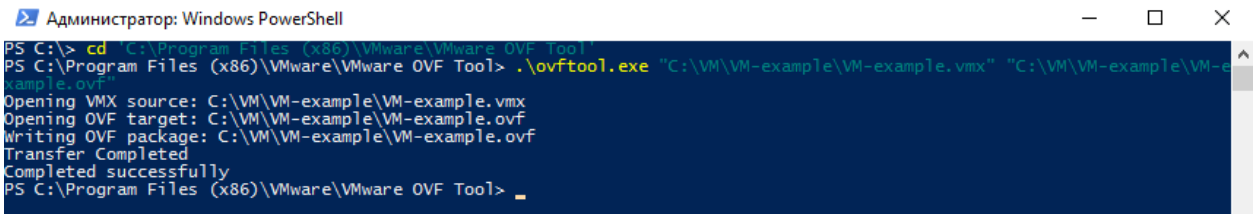

Рисунок 7.5. Ввод команды

После завершения конвертации у вас появится ovf файл с вашей виртуальной машиной.

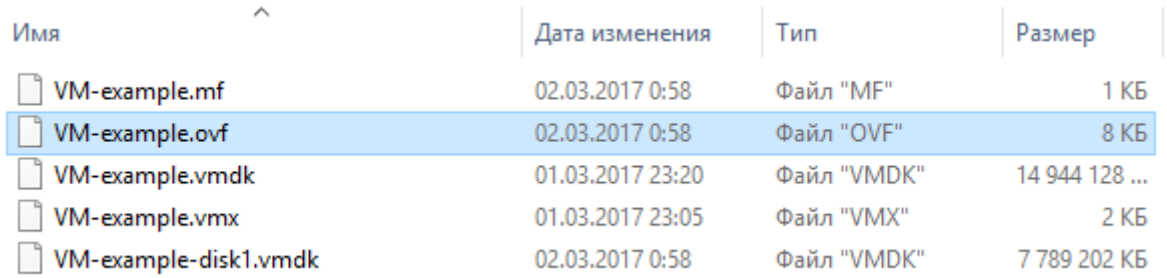

Рисунок 7.6. Файл с виртуальной машиной

7.2.5. Установка и настройка VNC-сервера

VNC (Virtual Network Computing) представляет собой клиент-серверный протокол, который позволяет клиентскому компьютеру (под управлением клиента VNC) подключаться и управлять удаленным компьютером (работающим на сервере VNC). Программное обеспечение было разработано исследовательской лабораторией Olivetti & Oracle в Кембридже, Великобритания, и его исходный код по-прежнему актуален и доступен под лицензией General Public License (GNU).

В зависимости от программного обеспечения сервера, VNC клиент будет подключаться к активному рабочему столу (например, как программы TeamViewer или AnyDesk) или к автономному виртуальному рабочему столу (как протокол удаленного рабочего стола Windows RDP). Последний, возможно, более мощный и безопасный, особенно если вам необходимо управлять серверным компьютером, потому что каждый сеанс будет уникальной средой, настроенной с разрешениями и правами подключаемого пользователя. Вот почему мы выбираем TigerVNC, который выполняет запуск параллельных сеансов среды рабочего стола компьютера (GNOME, KDE или другой GUI): это значит, что виртуальный рабочий стол будет создан для каждого отдельного подключения — это именно то, чего мы хотим.

Теперь давайте разберемся, как мы можем установить и настроить TigerVNC в CentOS. Первое, что нужно сделать, это установить программу TigerVNC Server, открыв сеанс терминала и введя следующую команду с правами root:

```
$ sudo yum install tigervnc-server
```

Сразу после этого вам необходимо создать отдельного пользователя VNC, с которого и будет осуществляться подключение (с выделенным паролем). Для этого введите следующее:

```
$ sudo adduser vncuser
$ sudo passwd vncuser
```

Никогда не делайте этого от имени root-пользователя — это создаст серьезную угрозу безопасности вашей системы. Лучшим решением будет оставить пользователя root без доступа к VNC и настроить выделенную учетную запись с ограниченными правами.

После того как вы создали vncuser и задали пароль для входа в систему, вам также необходимо установить уникальный пароль VNC для этого пользователя. Это можно сделать с помощью следующей команды:

```
$ su - vncuser
$ vncpasswd
```

(первая строка может быть опущена, если мы уже вошли под пользователем vncuser заранее).

Следующее, что нужно сделать, это создать файл конфигурации VNC для vncuser. Самый быстрый способ сделать это — скопировать файл общего шаблона VNC, расположенный в папке /lib/systemd/system/, — и затем изменить его:

```
$ cp /lib/systemd/system/vncserver@.service
/etc/systemd/system/vncserver@:1.service
```

Цифра «1», которую мы добавили в новое имя файла, — это номер дисплея, который будет использоваться для этого конкретного экземпляра службы. Это важно знать, потому что он также определит порт TCP, который будет использоваться нашим VNC-сервером, равным 5900 + номер дисплея. Первый будет 5901, затем 5902 и так далее.

Сразу после копирования вам нужно отредактировать новый файл с помощью Vi, Nano или другого текстового редактора, и заменить [USER] на имя пользователя, созданного недавно (в нашем случае vncuser). Вот как файл должен выглядеть после обновления (за исключением длинной закомментированной части в начале):

```
[Unit]
Description=Remote desktop service (VNC)
After=syslog.target network.target
[Service]Type=forking
# Clean any existing files in /tmp/.X11-unix environment
ExecStartPre=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1
|| :'
ExecStart=/usr/sbin/runuser -l vncuser -c "/usr/bin/vncserver %i -
geometry 1280x720"
PIDFile=/home/vncuser/.vnc/%H%i.pid
ExecStop=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 ||
:'
[Install]
WantedBy=multi-user.target
```

Листинг 7.3. Содержимое VNC файла

Как только вы это сделаете, можете перезагрузить демон VNC и запустить vncserver@1 с помощью следующих команд:

```
$ systemctl daemon-reload
$ systemctl start vncserver@:1
```

Прежде чем продолжить, проверьте, запущена ли служба, введя команду systemctl status:

```
$ systemctl status vncserver@:1
```

и создайте символическую ссылку, чтобы она всегда выполнялась при запуске системы, с помощью следующей команды:

```
$ systemctl enable vncserver@:1
```

Еще одна проверка, которую вы можете выполнить, прежде чем пытаться подключиться к серверу, — это посмотреть на активные сетевые сокеты с помощью команды ss: если все работает правильно, вы должны увидеть, что VNC-сервер работает и использует порт TCP 5901. Выполните команду:

```
$ ss -tulpn| grep vnc
```

Результат должен быть следующим:

```
tcp LISTEN 0 5 *:5901 *:* users:(("Xvnc",pid=38344,fd=9))
tcp LISTEN 0 128 *:6001 *:* users:(("Xvnc",pid=38344,fd=6))
tcp LISTEN 0 5 :::5901 :::* users:(("Xvnc",pid=38344,fd=10))
tcp LISTEN 0 128 :::6001 :::* users:(("Xvnc",pid=38344,fd=5))
```

Листинг 7.4. Вывод команды ss –tulpn

Поскольку наша служба VNC прослушивает порт TCP 5901, вы должны быть уверены, что такой порт открыт и доступен внешним клиентам. Поэтому если у вас установлен брандмауэр, вы должны создать соответствующее правило, позволяющее подключаться клиентам VNC. Сколько открыть портов, будет зависеть от того, сколько экземпляров VNC-серверов вам понадобится. В нашем сценарии мы сделали только 1, поэтому вам нужно открыть только первый выделенный порт VNC: TCP 5901, как мы уже и говорили. Команда, чтобы открыть этот порт на Firewalld:

```
# firewall-cmd --add-port = 5901 / tcp
# firewall-cmd --add-port = 5901 / tcp –permanent
```

Не будет лишним упомянуть, что вы также можете ограничить этот порт для определенных групп, IP-адресов, сетевых карт или других простых, или сложных правил брандмауэра.

Если у вас уже есть GNOME, KDE или другие установленные среды рабочего стола, вы можете пропустить этот шаг. В противном случае вам необходимо установить один из них: сервер TigerVNC будет запускать параллельный экземпляр данной среды рабочего стола для каждого сеанса входа в систему, что означает, что у нас должен быть хотя бы один GUI.

Если вы хотите действительно отличный, но тяжеловесный интерфейс GNOME, введите следующее:

```
$ sudo yum groupinstall "GNOME Desktop"
```

Если вам нужна легкая альтернатива, мы можем предложить Xfce — бесплатную среду с открытым исходным кодом для Unix-подобных платформ, которая прекрасно работает с TigerVNC. GNOME также отлично работает, но он

довольно ресурсоемкий: если вы хотите сэкономить ресурсы на своей серверной машине, Xfce, возможно, будет лучшим выбором. Чтобы установить его, введите следующее:

```
$ yum install epel-release
$ yum groupinstall xfce
```

Если вы решили использовать Xfce, вам также потребуется изменить файл, который выполнялся при запуске сеанса VNC. Для этого отредактируйте файл:

```
/home/<user>/.vnc/xstartup
```

и измените запись exec (обычно строка 4) с etc/X11/xinit/xinitrc на startxfce4, как показано ниже:

```
#!/bin/sh
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
exec startxfce4
vncserver -kill $DISPLAY
```

Листинг 7.5. Запись exec

Теперь, когда все настроено, мы можем попытаться подключиться к нашей службе VNC с помощью клиента VNC, такого как TightVNC, UltraVNC или RealVNC, и посмотреть, что произойдет. Не забудьте указать порт TCP 5901.

ЗАКЛЮЧЕНИЕ

В учебном пособии даны понятийный аппарат и основы администрирования подсистем обеспечения информационной безопасности. Рассмотрены основные средства и способы обеспечения информационной безопасности в автоматизированных системах. Отражены концепции и принципы администрирования подсистем обеспечения информационной безопасности.

Также в пособии приведены и глубоко рассмотрены следующие вопросы:

– общие положения администрирования подсистем обеспечения информационной безопасности;

– организационные аспекты управления информационной безопасностью;

– администрирование элементов обеспечения информационной безопасности компьютерных сетей;

– администрирование элементов обеспечения информационной безопасности рабочих станций на основе операционных систем Windows;

– администрирование элементов обеспечения информационной безопасности рабочих станций на основе операционных систем Linux;

– администрирование элементов обеспечения информационной безопасности серверов на основе операционных систем Windows;

– администрирование элементов обеспечения информационной безопасности серверов на основе операционных систем Linux;

– администрирование элементов обеспечения информационной безопасности при использовании технологий виртуализации.

Данное учебное пособие может быть также использовано на занятиях в системе служебной подготовки и для профессиональной подготовки сотрудников подразделений информационных технологий, связи и защиты информации территориальных органов МВД России.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ Р ИСО/МЭК 27000-2012. Национальный стандарт Российской Федерации. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Общий обзор и терминология. // СПС «КонсультантПлюс». – Режим доступа: http://www.consultant.ru.

2. ГОСТ Р ИСО/МЭК 27001-2006. Национальный стандарт Российской Федерации. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. // СПС «КонсультантПлюс». – Режим доступа: http://www.consultant.ru.

3. Carvalho L. Windows Server 2012 Hyper-V cookbook: over 50 simple but incredibly effective recipes for mastering the administration of Windows Server Hyper-V / L. Carvalho, Birmingham: Packt Publ, 2012. 285 c.

4. Clifton L. Mastering Microsoft Exchange Server 2016 / L. Clifton, Indianapolis, Indiana: Sybex, a Wiley brand, 2016. 782 c.

5. Krause J. Mastering Microsoft Windows Server 2016 a comprehensive and practical guide to Windows Server 2016 / J. Krause, Birmingham: Packt, 2016.

6. Krause J. Windows server 2016 cookbook / J. Krause, 494 c.

7. Kurose J. F., Ross K. W. Computer networking: a top-down approach / J. F. Kurose, K. W. Ross, 6th ed-е изд., Boston: Pearson, 2013. 862 c.

8. Lee T. Windows Server 2016 Automation with PowerShell Cookbook - Second Edition / T. Lee, 2017.

9. Mackin J. C. Exam ref 70-410 installing and configuring windows server threshold / J. C. Mackin, Redmond, WA: Microsoft Press, 2015.

10. Savill J. Mastering Windows server 2016 Hyper-V / J. Savill, Indianapolis, Indiana: John Wiley & Sons, 2017. 626 c.

11. Бреснахэн К., Блум Р. Linux на практике. - СПб.: Питер, 2017. - 384 с.: ил. - (Серия «Для про-фессионалов»).

12. Минаси, Марк, Грин, Кевин, Бус, Кристиан, Батлер, Роберт, и др. Windows Server 2012 R2. Полное руководство. Том l: установка и конфигурирование сервера, сети, DNS, Active Directory и общего доступа к данным и принтерам.: Пер. с англ. - М.: ООО «И.Д. Вильямс». 2015. - 960 с.: ил. - Парал. тит. англ.

13. Минаси, Марк, Грин, Кевин, Бус, Кристиан, Батлер, Роберт, и др. Windows Server 2012 R2. Полное руководство. Том 2: дистанционное администрирование, установка среды с несколькими доменами, виртуализация, мониторинг и обслуживание сервера.: Пер. с англ. - М.: ООО «И.Д. Вильямс». 2015. - 864 с.: ил. - Парал. тит. англ.

14. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. – СПБ.: Питер, 2016. – 992 с.: ил. – (Серия «Учебник для вузов»).

15. Основы сетевой инфраструктуры Windows Server 2008 2008. С. 139.

16. Роббинс, Арнольд. Bash. Карманный справочник системного администратора, 2-е изд.: Пер. с англ. – СПб.: ООО «Альфа-книга», 2017. – 152 с.: ил. – Парал. тит. англ.

17. Станек У.Р. Microsoft Windows Server® 2012. Справочник администратора: Пер. с англ. — М.: Издательство «Русская редакция»; СПб.: «БХВ-Петербург», 2014. — 688 с.: ил. — (Справочник администратора).

18. У. Ричард Стивенс, Стивен А. Раго. Unix. Профессиональное программирование. 3-е изд. – СПБ.: Питер, 2018. – 944 с.: ил. – (Серия «Для профессионалов»).

19. Холмс Д., Реет Н., Реет Д. Настройка Active Directory®. Windows Server® 2008. Учебный курс Microsoft / Пер. с англ. — М.: Издательство «Русская редакция», 2011. — 960 стр.: ил.

20. Шоттс У. Командная строка Linux. Полное руководство. — СПб.: Питер, 2017. — 480 с.: ил. — (Серия «Для профессионалов»).

Учебное издание

АДМИНИСТРИРОВАНИЕ ПОДСИСТЕМ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Учебное пособие

В авторской редакции

ISBN 978-5-9266-1731-0

Подписано в печать 22.11.2021.
Авт. л. 9,5. Заказ 181.

Краснодарский университет МВД России.
350005, г. Краснодар, ул. Ярославская, 128.