ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Северо-Кавказский филиал ордена Трудового Красного Знамени
федерального государственного бюджетного образовательного учреждения
высшего образования
«Московский технический университет связи и информатики»

Манин А.А.

Методы и средства защиты компьютерной информации

Учебное пособие

Ростов-на-Дону

2019 г.

Манин А.А. Методы и средства защиты компьютерной информации. Учебное пособие. – Ростов-на-Дону, 2019.

В учебном пособии рассмотрены основные принципы защиты от несанкционированного доступа к информации, циркулирующей в компьютерных сетях, построенных на базе стека протоколов TCP/IP. Рассмотрены основные модели политик сетевой безопасности, особенности обеспечения защищенного доступа к сетевому оборудованию. Особенности межсетевого экранирования рассмотрены на примере возможностей, встроенных в операционные системы сетевых устройств Cisco Systems. Рассмотрены дополнительные средства защиты – виртуальные частные сети, технология NAT, средства защиты от атак на канальном уровне.

Пособие предназначено для студентов, обучающихся по направлению 11.03.02 «Инфокоммуникационные технологии и системы связи» (бакалавриат), профиль «Защищенные системы и сети связи», а также по направлению 09.03.01 «Информатика и вычислительная техника» (бакалавриат), профиль «Программное обеспечение и интеллектуальные системы».

Рецензенты: доктор техн. наук, доцент Елисеев А.В.

доктор техн. наук, доцент Погорелов В.А.

СОДЕРЖАНИЕ

Введение

Дисциплина «Методы и средства защиты компьютерной информации» направлена на формирования у студента знаний и навыков в области обеспечения защиты информации, передаваемой по сети, от несанкционированного доступа.

В настоящее время проблема защиты таких сетей особенно обострилась с широким распространением локальных и, особенно, глобальных сетей связи. Это связано, прежде всего, с тем, что подавляющее большинство этих сетей используют стек протоколов TCP/IP, который изначально являлся незащищенным. Протоколы, включенные в стек TCP/IP, предназначались только для обеспечения доставки информации, и никак не учитывали ни возможные угрозы, ни возможный ущерб.

Необходимо также отметить, что зачастую ущерб наносится не из-за действий злоумышленников, а из-за элементарных ошибок пользователей, которые случайно модифицируют или удаляют жизненно важные данные. В связи с этим, помимо контроля доступа, необходимым элементом защиты информации в компьютерных сетях является разграничение полномочий пользователей.

В компьютерных сетях при организации контроля доступа и разграничения полномочий пользователей могут использоваться встроенные средства сетевых операционных систем, а также специализированные аппаратно-программные средства. Эти средства могут использовать дополнительные протоколы, включенные в стек TCP/IP для обеспечения защиты, и дополнительные функции, не стандартизированные в протоколах и предназначенные для обеспечения безопасности. Соответственно, для построения защищенной сети необходимо знание этих протоколов и функций, а также умение конфигурировать сетевое оборудование.

Учебное пособие разбито на главы. В первой главе рассматриваются основные модели политик сетевой безопасности. Эти модели вынесены в начало курса, так как все технические методы и средства защиты,

рассматриваемые в последующих главах, направлены на реализацию политик безопасности.

Во второй и третьей главах рассматриваются вопросы обеспечения защиты от несанкционированного доступа к сетевому оборудованию. Вторая глава посвящена защите сетевых устройств с использованием локальной базы пользователей, а третья – с использованием AAA-серверов.

В четвертой главе рассмотрено межсетевое экранирование. Практическое применение межсетевых экранов иллюстрируется на базе возможностей, встроенных в операционную систему устройств Cisco Systems.

В пятой главе рассмотрены дополнительные средства защиты сетей – использование технологии NAT, виртуальных частных сетей, встроенных средств защиты от атак на наиболее распространенные сетевые протоколы.

В шестой главе рассматриваются вопросы защиты конфиденциальных данных, передаваемых по общедоступным сетям, на основе технологий виртуальных частных сетей (VPN).

Весь материал учебного пособия имеет практическую направленность и содержит примеры конфигурирования сетевых устройств. Поэтому после изучения теоретического материала каждой главы студенты выполняют индивидуальные практические задания. По результатам выполнения этих заданий рассчитываются баллы студентов в рамках модульно-рейтинговой системы.

1. Политика безопасности

1.1 Общие понятия

Прежде чем ввести в рассмотрение понятие политики безопасности, необходимо определиться, что именно мы понимаем под термином «безопасность».

В нашем случае мы имеем дело с компьютерной информацией, поэтому дальше речь пойдет об информационной безопасности.

Информационная безопасность (ИБ) – это состояние информационной системы, при котором она наименее восприимчива к вмешательству и нанесению ущерба со стороны третьих лиц (нарушителей).

В этом курсе мы будем вести речь не об информационной, а о компьютерной безопасности. Это понятие более узкое, и не включает в себя такие аспекты, как организационные меры, технические средства защиты от прослушивания, и т.д.

Информация считается защищенной, если соблюдаются три главных свойства.

Первое – **целостность** – предполагает обеспечение достоверности и корректного отображения охраняемых данных, независимо от того, какие системы безопасности и приемы защиты используются. Обработка данных не должна нарушаться, а пользователи системы, которые работают с защищаемыми файлами, не должны сталкиваться с несанкционированной модификацией или уничтожением ресурсов, сбоями в работе ПО.

Второе – **конфиденциальность** – означает, что доступ к просмотру и редактированию данных предоставляется исключительно авторизованным пользователям системы защиты.

Третье – **доступность** – подразумевает, что все авторизованные пользователи должны иметь доступ к конфиденциальной информации.

Достаточно нарушить одно из свойств защищенной информации, чтобы использование системы стало бессмысленным.

В связи с этим вводят понятия: уязвимость, атака, нарушитель.

Уязвимость – недостаток в системе, используя который, можно намеренно нарушить её целостность и вызвать неправильную работу.

Атака – реализация уязвимости с целью нарушения информационной безопасности.

Различают следующие типы атак:

- атаки доступа;
- атаки модификации;
- атаки на отказ в обслуживании;
- атаки на отказ от обязательств.

Атака доступа – это попытка доступа нарушителем к информации, для просмотра которой у него нет полномочий. Направлена на нарушение конфиденциальности.

Атака модификации – это неправомочная попытка изменения информации. Направлена на нарушение целостности.

Атаки на отказ в обслуживании направлены на обеспечение невозможности доступа легальных пользователей к информации. Направлены на нарушение доступности.

Атаки на отказ от обязательств можно рассматривать как разновидность атак модификации. Например, подменив сетевой адрес, нарушитель может выдать себя за другого пользователя или отрицать факт какого-либо действия.

Под политикой безопасности понимается совокупность норм и правил, регламентирующих процесс обработки информации, выполнение которых обеспечивает защиту от определенного множества угроз и составляет необходимое условие безопасности системы. Формальное выражение политики безопасности называют моделью безопасности.

В нашем случае под системой мы понимаем сеть связи, поэтому при дальнейшем рассмотрении будем в качестве компонент системы понимать сетевые ресурсы.

С сетевой точки зрения модель безопасности представляет собой набор правил разграничения доступа к сетевым ресурсам. В связи с этим вводятся понятия субъекта, объекта и операции.

Субъект – любая сущность, способная инициировать операции над объектами. Часто к субъектам относят не только пользователей, но и процессы, сгенерированные различными приложениями. Однако с точки зрения защиты сетей логичнее к субъектам отнести именно пользователей, от имени которых выполняются процессы. Причем здесь имеется в виду не конкретный человек, а «логический» пользователь, осуществивший вход в систему.

Объект – сетевой ресурс, к которому может быть обеспечен доступ объекта. К объектам относятся как аппаратные средства – маршрутизаторы, коммутаторы, память – так и программные – файлы, прикладные программы, и т.д.

Операции – действия, которые субъекты могут выполнять над объектами. Вид операций зависит от вида объекта. Например, для файлов могут существовать операции «чтение», «запись», для маршрутизаторов – выполнение определенных команд конфигурирования, и т.д.

В самом общем виде политика безопасности должна содержать набор правил, описывающих права доступа субъектов к объектам, причем если определенный субъект имеет доступ к объекту, необходимо указать, какие именно операции разрешены.

Очевидно, что для обеспечения выполнения требований политики безопасности каждый субъект должен быть идентифицирован.

Идентификация заключается в том, что субъект сообщает о себе системе некоторую идентифицирующую информацию. Для того, чтобы установить, что пользователь именно тот, за которого себя выдает, используется процедура аутентификации.

Аутентификация (Authentication) субъекта заключается в том, что помимо идентифицирующей информации пользователь сообщает системе

дополнительную информацию, подтверждающую, что он именно тот, за кого себя выдает. Вид такой информации зависит от способа аутентификации (будут рассмотрены ниже), например, это может быть пароль, хэш-функция, цифровая подпись, токен, биометрические данные, и т.д.

После аутентификации субъекта проводится его **авторизация** (Authorization) – определение тех операций, которые ему разрешены над объектами. Эти операции должны быть строго определены в политике безопасности.

Наконец, для анализа проблем сетевой безопасности необходим **учет** всех операций субъектов над объектами (Accounting).

Процедуры аутентификации, авторизации и учета объединяют единым термином – AAA (Authentication, Authorization, Accounting). Устройства, реализующие в сети процедуры AAA, называются AAA-серверами.

Особо следует отметить, что необходимыми условиями корректной работы политики безопасности являются:

1. В любой момент времени любой объект и субъект должны быть идентифицированы и аутентифицированы.

2. В защищенной системе (в том числе в сети связи) должен присутствовать субъект с соответствующим ему объектом, который управляет доступом и осуществляет контроль доступа субъектов к объектам. Такой субъект называется **монитором безопасности.**

3. Для осуществления своих функций монитор безопасности (как субъект) должен обладать определенной информацией, которая содержится в особом объекте.

Таким образом, монитор безопасности – это совокупность аппаратных и программных средств, реализующих политику безопасности. Часто используется английское обозначение – Trusted Computing Base (TCB).

К монитору безопасности предъявляются следующие требования.

1. Полнота. Монитор безопасности должен вызываться при каждом обращении любого субъекта к любому объекту, и не должно существовать никаких способов его обхода.

2. Изолированность. Монитор безопасности должен быть защищен от отслеживания и перехвата него работы.

3. Верифицируемость. Монитор безопасности должен быть проверяемым на предмет выполнения своих функций.

4. Непрерывность. Монитор безопасности должен функционировать при любых ситуациях, в том числе аварийных.

Из условия 3 корректной работы политики безопасности следует, что ассоциируемый с монитором безопасности объект, содержащий информацию о правилах разграничения доступа, является наиболее критическим информационным ресурсом с точки зрения безопасности. Кроме того, в защищенной системе должен присутствовать особый доверенный пользователь (администратор) субъекты которого имеют доступ к ассоциированному с монитором безопасности объекту.

1.2 Типы политик безопасности

1.2.1 Дискреционная модель политики безопасности

Как указывалось выше, ключевым понятием политики безопасности является понятие «доступ».

Основой дискреционной (избирательной) политики безопасности является дискреционное управление доступом (Discretionary Access Control – DAC), которое определяется двумя свойствами:

1. Все субъекты и объекты должны быть идентифицированы.

2. Права доступа субъекта к объекту определяются на основании заданного правила.

Чаще всего дискреционная политика формализуется в виде матрицы доступа, в которой в первом столбце указываются субъекты, в первой строке

– объекты, а на их пересечении – разрешенные операции. Приведем пример дискреционной модели, реализуемой в большинстве операционных систем.

Предположим, что имеется три пользователя (субъекты) и пять объектов – файл 1, файл 2, файл 3, CD-ROM и HDD. При этом операции, которые можно производить над объектами: W (запись), R (чтение), M (модифицировать), E (исполнять), C (создавать), F (полный доступ). Матрица доступа представлена в таблице 1.1.

Таблица 1.1 – Матрица доступа

	Файл1	Файл 2	Файл 3	CD-ROM	HDD
Пользователь1	F	F	F	F	F
Пользователь2	W, M	R	-	-	E
Пользователь3	-	W, M	-	C, E	-

В операционных системах, в отличие от сетей, практически всегда имеется владелец объекта, как правило, тот субъект, который его создал. Соответственно, владелец может делегировать другим субъектам права доступа. Пример прав доступа операционной системы Windows показан на рисунке 1.1.

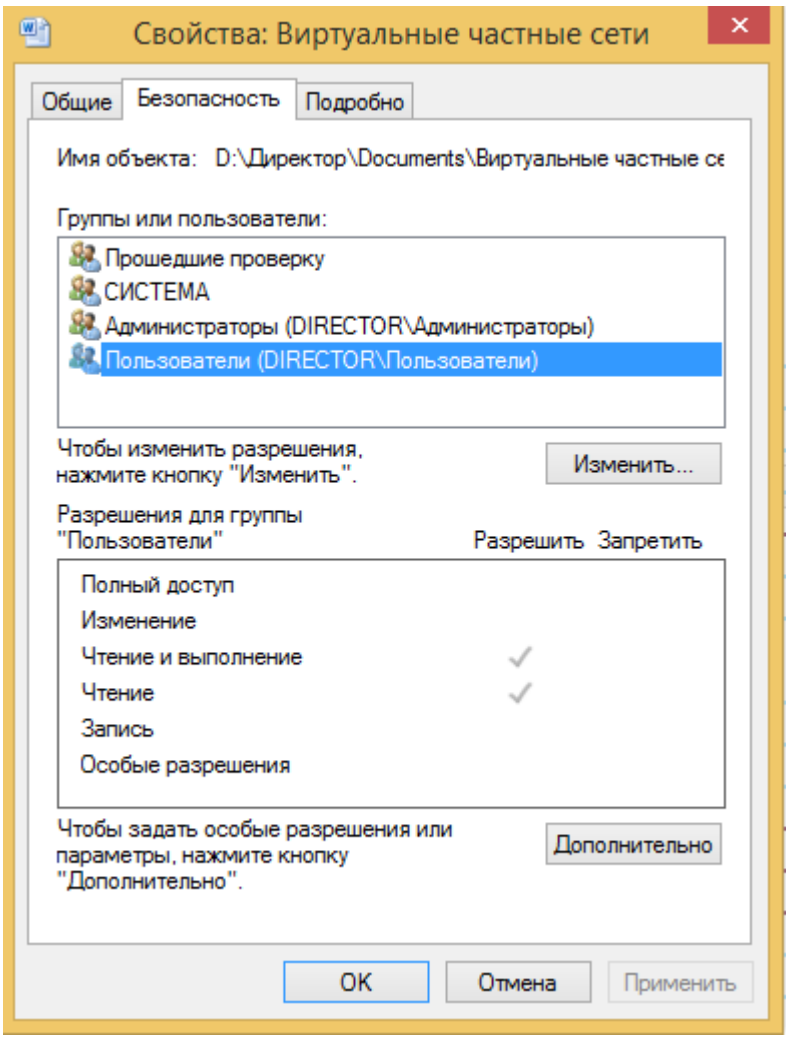


Рисунок 1.1 – Пример прав доступа к файлу в ОС Windows

В сетевой инфраструктуре объекты, как правило, создаются не пользователями, а администраторами сети, которые, являясь владельцем сетевого ресурса, могут предоставить права доступа тем или иным субъектам. Соответственно, у администратора как владельца объекта имеется полный доступ к сетевым ресурсам (как у пользователя 1 в таблице 1.1).

Рассмотрим сетевую структуру, показанную на рисунке 1.2.

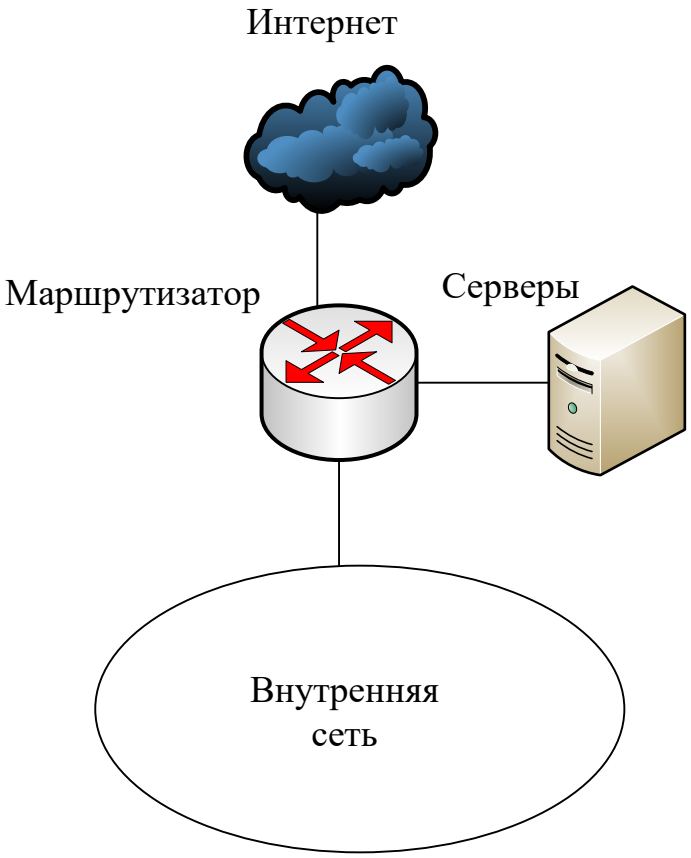


Рисунок 1.2 – Структура сети

В этом случае администратору необходимо определить те права доступа, которые необходимы сотрудникам для своей работы. Однако теперь в качестве объектов выступают не файлы или память ПК, а сетевые ресурсы, расположенные на серверах и в Интернете. Поэтому при назначении прав доступа администратор должен указать каким субъектам (пользователям) должен быть предоставлен к каким ресурсам (объектам) и с использованием каких сетевых протоколов. Пример матрицы доступа представлен в таблице 1.2.

Таблица 1.2 – Матрица доступа

	Веб-сервер	Почтовый сервер	Интернет
Администратор	SSH, Telnet, HTTP,HTTPS	SSH, Telnet, POP,SMTP	HTTP,HTTPS
Руководство	HTTP,HTTPS	POP,SMTP	HTTP,HTTPS
Персонал	-	POP,SMTP	-

Таблица 1.2 лишь показывает саму идею и сильно упрощена, реальные матрицы намного сложнее.

Достоинством дискреционной модели является достаточно простая реализация.

К недостаткам можно отнести статичность политики, то есть при любом изменении структуры сети, состава пользователей, набора сетевых ресурсов необходимо вносить изменения в матрицу доступа, и переконфигурировать оборудование.

1.2.2 Мандатная модель политики безопасности

Политика мандатного доступа является примером использования технологий, наработанных во внекомпьютерной сфере. Ярким примером таких технологий является организация секретного делопроизводства и документооборота.

Основным принципом политики мандатного доступа является назначение всем участникам процесса (субъектам) специальной метки, называемой **уровнем безопасности**. Например, могут существовать уровни секретно, совершенно секретно, и т.д. Все уровни безопасности должны быть упорядочены в иерархической структуре, то есть, например, уровень совершенно секретно всегда находится выше уровня секретно. При этом каждому объекту присваивается уровень безопасности в зависимости от содержащейся в нем информации, и каждому субъекту присваивается свой уровень в зависимости от его прав. Субъект имеет право совершать операции только над теми объектами, уровень которых не выше его собственного. Например, сотрудник, имеющий право работать с документами с грифом «секретно», не имеет доступа к документам с грифом «совершенно секретно».

Приведем пример. Пусть в системе имеется следующий набор **уровней безопасности**:

- Top Secret

- Secret

- Confidential

- Non Confidential

Множество **субъектов**:

- Администратор

- Пользователь 1

- Пользователь 2

- Гость

Множество **объектов**:

- Файл 1

- Файл 2

- Файл 3

- CD ROM

- HDD

Соответствие субъектов уровням безопасности иллюстрируется таблицей 1.3.

Таблица 1.3

Администратор	Пользователь 1	Пользователь 2	Гость
Top Secret	Secret	Confidential	Non Confidential

Соответствие объектов уровням безопасности иллюстрируется таблицей 1.4.

Таблица 1.4

HDD	CD ROM	Файлы 1, 2	Файл 3
Top Secret	Secret	Confidential	Non Confidential

В этом случае монитор безопасности при попытке доступа субъекта к объекту проверяет уровень безопасности и либо предоставляет доступ, либо отказывает в нем.

Однако у такой политики есть существенный недостаток. Например, если субъект Гость завербует субъекта Пользователь 2, и попросит его записать информацию, содержащуюся в файле 1, в файл 2, формально политика безопасности нарушена не будет. Однако субъект Гость получит доступ к информации с уровнем Confidential.

Поэтому мандатная политика дополняется принципом Бела-Лападулы, который формулируется следующим образом.

Для мандатной политики обязательным является выполнение следующих правил:

1. No Read Up (NRU) – нет чтения вверх: субъект имеет право читать только те документы, уровень безопасности которых не выше его уровня;

2. No Write Down (NWD) – нет записи вниз: субъект имеет право заносить информацию только в не документы, уровень безопасности которых не ниже его уровня.

1.2.3 Ролевая политика безопасности

В основе рассмотренных выше моделей политики безопасности лежат взаимоотношения между субъектами и объектами. Эти взаимоотношения определяются либо внешним фактором (дискреционный доступ), либо внутренним (мандатный доступ).

Однако в реальной жизни и работе пользователи выполняют различные задачи не от своего личного имени, а в рамках некоторой должности с соответствующими функциональными обязанностями. Такую должность можно трактовать как некоторую **роль**, которая представляет собой некоторую абстрактную сущность с определенными правами и полномочиями.

Ролью называется действующая в системе абстрактная сущность, обладающая некоторым объемом полномочий, необходимых для выполнения определенных обязанностей пользователя.

Введение ролей приводит к двухэтапной организации системы разграничения доступа:

1. Создание ролей и определение их прав доступа к объектам.

2. Назначение ролей субъектам (пользователям).

При этом определение прав доступа ролей к объектам производится по принципу наименьших полномочий. Иначе говоря, каждая роль должна иметь только тот минимальный набор прав доступа, который необходим для выполнения их функциональных обязанностей.

В ролевой модели вводятся следующие множества:

- множество пользователей;
- множество ролей;
- множество полномочий (таблица прав доступа);
- множество сеансов.

2 Защита от несанкционированного доступа к сетевому оборудованию

2.1 Защита консольного доступа

Для реализации любой из рассмотренных выше моделей политики безопасности необходимы определенные действия с сетевым оборудованием. Эти действия производятся путем конфигурирования, которое обычно отличается у оборудования различных производителей. Однако общий принцип остается неизменным, поэтому здесь будем рассматривать приемы конфигурирования оборудования Cisco Systems и Huawei Technologies, главным образом потому, что данные производители предоставляют для обучения свои сетевые симуляторы – Cisco Packet Tracer и eNSP.

Как известно, при начальном конфигурировании оборудования (коммутаторов и маршрутизаторов) необходимо подключиться к ним с помощью консольного порта (рисунок 2.1).

Для подключения по консоли необходимо использовать программу эмуляции терминала. В ранних версиях Windows использовалась программа Hyper Terminal, в настоящее время широко используются сторонние программы, например, PuTTy.

- Между последовательным портом (COM) и консольным интерфейсом маршрутизатора/коммутатора установлено физическое соединение.

Рисунок 2.1 – Иллюстрация консольного доступа

При успешном подключении отобразится приглашение операционной системы устройства. Вид приглашения зависит от типа устройства и от того режима, в котором мы находимся. Режимы оборудования Cisco показаны в таблице 2.1.

Таблица 2.1 – Режимы оборудования Cisco

Название режима	Приглашение (Prompt)	Описание
User EXEC Mode	Switch>	Пользовательский режим
Privileged EXEC Mode	Switch #	Привилегированный режим
Global Configuration Mode	Switch (config)#	Режим глобального конфигурирования

В таблице 2.1 в первом столбце представлены названия режимов, во втором – приглашение, отображаемое в командной строке, в третьем – описание.

Пользовательский режим (user mode) используется для просмотра состояния устройства, а также для перехода в привилегированный режим (privileged mode). Никаких изменений в конфигурационном файле, в том числе удаления и сохранения текущей конфигурации, в пользовательском режиме производиться не может. В этом режиме доступны только некоторые команды верификации **show**, т. е. команды просмотра состояния устройства.

Для перехода в привилегированный режим в устройствах Cisco используется команда **enable.** Так как привилегированный режим является потенциально опасным, рекомендуется защитить переход в этот режим паролем. Как известно [1], в устройствах Cisco существует несколько видов паролей. В частности, пароли **enable password** и **enable secret** как раз и обеспечивают авторизацию входа в привилегированный режим.

Данные пароли устанавливаются в режиме конфигурирования с использованием команд:

R1(config)#enable password manin

R1(config)#enable secret manin1

Первая команда устанавливает пароль типа **enable password** (для примера был выбран пароль **manin**), вторая – типа **enable secret** (пароль **manin1**).

Пароль типа **enable password** хранится на устройстве в незашифрованном виде, что делает его слабозащищенным. Например, если злоумышленник подключится по консоли, войдет в пользовательский режим и использует команду **show running-config**, он без труда сможет увидеть установленный пароль (рисунок 2.2).

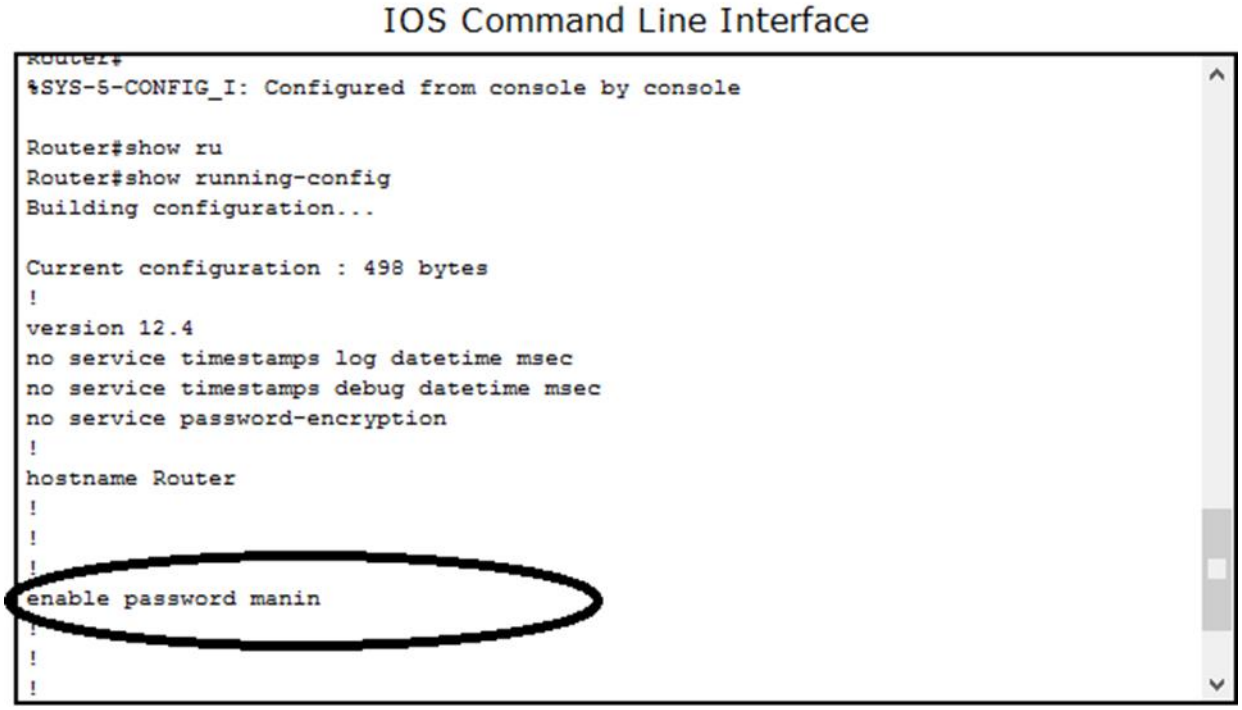


Рисунок 2.2 – Просмотр пароля

Самым очевидным решением этой проблемы является хранение пароля в зашифрованном виде. Для шифрования пароля можно использовать команду **service password-encryption**, выполняемую в режиме глобального конфигурирования. В этом случае при просмотре конфигурации пароль будет представлен в зашифрованном виде (рисунок 2.3).

Однако на практике не рекомендуется данный способ защиты перехода в привилегированный режим. Дело в том, что шифрование в этом случае производится с использованием алгоритма шифрования, основанного на операции XOR и достаточно легко поддающегося взлому. Цифра 7 перед зашифрованным паролем как раз и указывает на то, что используется этот слабо защищенный алгоритм шифрования.

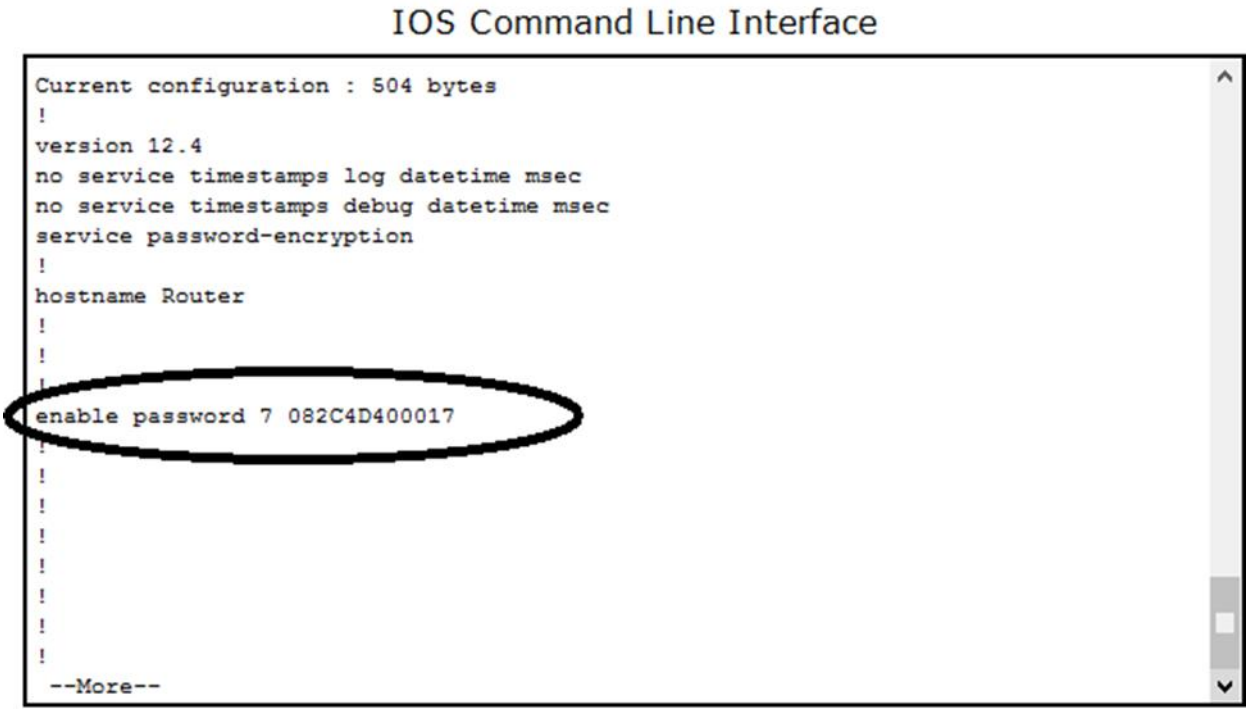


Рисунок 2.3 – Просмотр зашифрованного пароля

Пример расшифровки пароля, показанного на рисунке 2.3, с использованием одной из известных программ показан на рисунке 2.4.

Рисунок 2.4 – Расшифровка пароля

Второй тип пароля – **enable secret** – использует более мощный алгоритм шифрования, основанный на использовании хэш-функций (MD5). Создадим пароль **enable secret** и посмотрим его отображение в конфигурации устройства, рисунок 2.5.

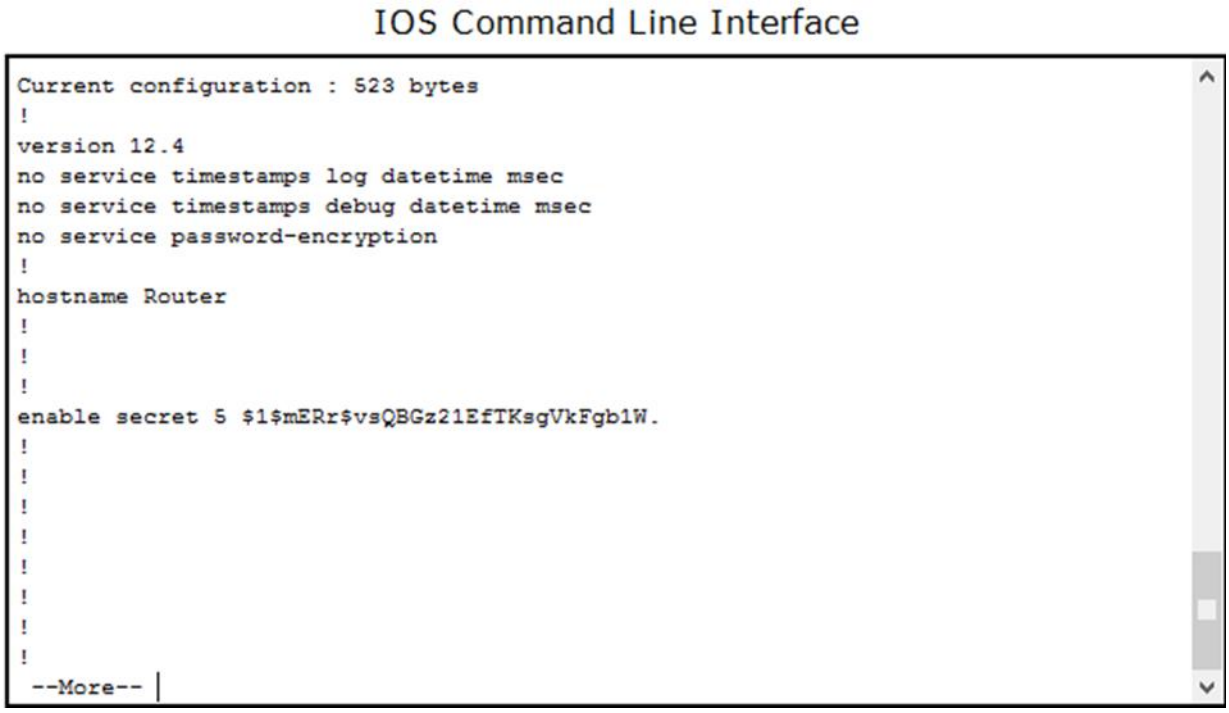


Рисунок 2.5 – Отображение пароля **enable secret**

Как видно из рисунка, теперь перед паролем стоит цифра 5, что и указывает на использование алгоритма MD5, и мы видим результат выполнения хэш-функции, которая, как известно, необратима [2].

Однако следует помнить, что в этом случае пароль можно подобрать. Существуют специальные таблицы, включающие в себя результаты выполнения хэш-функций для наиболее часто встречающихся паролей. В Интернете есть специализированные ресурсы, подбирающие пароли к хэш-функциям. Трудоемкость подбора пароля **enable secret** напрямую зависит от его сложности. Отсюда следует очевидный вывод – пароль должен быть достаточно сложным и нетривиальным.

Таким образом, с использованием пароля необходимо защитить привилегированный режим, соответственно, и режим конфигурирования.

Помимо защиты от несанкционированного доступа к привилегированному режиму, при конфигурировании оборудования необходимо обеспечить защиту и пользовательского режима. Как указывалось выше, при входе в пользовательский режим злоумышленник также может выполнить ряд команд и получить некоторые сведения об устройстве.

Для того, чтобы защитить не только привилегированный, но и пользовательский режим, используется аутентификация по учетным записям пользователей.

Как известно, при использовании учетной записи аутентификация проводится, как минимум, по двум параметрам – логину и паролю. Очевидно, что логины и пароли легальных пользователей должны где-то храниться. Для хранения учетных записей используется два подхода:

1. Хранение в локальной базе непосредственно на устройстве.

2. Хранение на специализированном сервере (AAA-сервере).

Наиболее надежным и удобным в эксплуатации способом является использование AAA-сервера (серверов). Однако в небольшой сети использование AAA-сервера обычно представляется нецелесообразным, поэтому рассмотрим сначала первый способ.

При создании базы пользователей необходимо знать, что устройства Cisco позволяют достаточно гибко управлять теми правами (привилегиями), которые предоставляются каждому конкретному пользователю. Имеется 16 уровней привилегий (от 0 до 15), при этом по умолчанию в устройстве Cisco настроены три уровня:

1. Уровень 0. Пользователь с этим уровнем может выполнять минимальный набор команд. На практике используется крайне редко.

2. Уровень 1. Соответствует пользовательскому режиму, то есть пользователь с этим уровнем может выполнять все команды, доступные в пользовательском режиме.

3. Уровень 15. Соответствует привилегированному режиму, то есть пользователь с этим уровнем может выполнять все команды, доступные в привилегированном режиме.

Уровни 2 – 14 можно настраивать, то есть, если какому-либо пользователю присваивается уровень из этого диапазона, можно в явном виде указать, какие команды привилегированного режима будут ему доступны. Если учесть, что число работников, обслуживающих сеть, обычно невелико, такой подход позволяет каждому из работников предоставить только те права, которые необходимы ему для работы.

Создание учетной записи производится в режиме глобального конфигурирования с использованием команды

Router1(config)#username <логин> privilege <уровень> password/secret <пароль>

В команде используется параметр либо **password**, либо **secret**, разница между ними была рассмотрена выше. Соответственно, более предпочтительным является использование параметра **secret**.

В случае, если создается учетная запись с привилегиями уровней 2 – 14, доступные на этом уровне команды указываются в явном виде:

Router1(config)#privilege exec level <уровень> <команда>

Рассмотрим процесс создания локальной базы на конкретном примере. Предположим, что в организации имеется три администратора (назовем их admin1, admin2 и admin3). Admin1 является главным, ему доступны все команды (уровень 15). Admin2 имеет доступ к командам **show running-config** и **ping**, admin3 – к командам **show ip route**, **ping** и **traceroute**.

Создание локальной базы пользователей иллюстрируется рисунком 2.6.

После этого необходимо войти в режим конфигурирования консольного порта и указать, что вход необходимо производить с использованием локальной базы, рисунок 2.7.

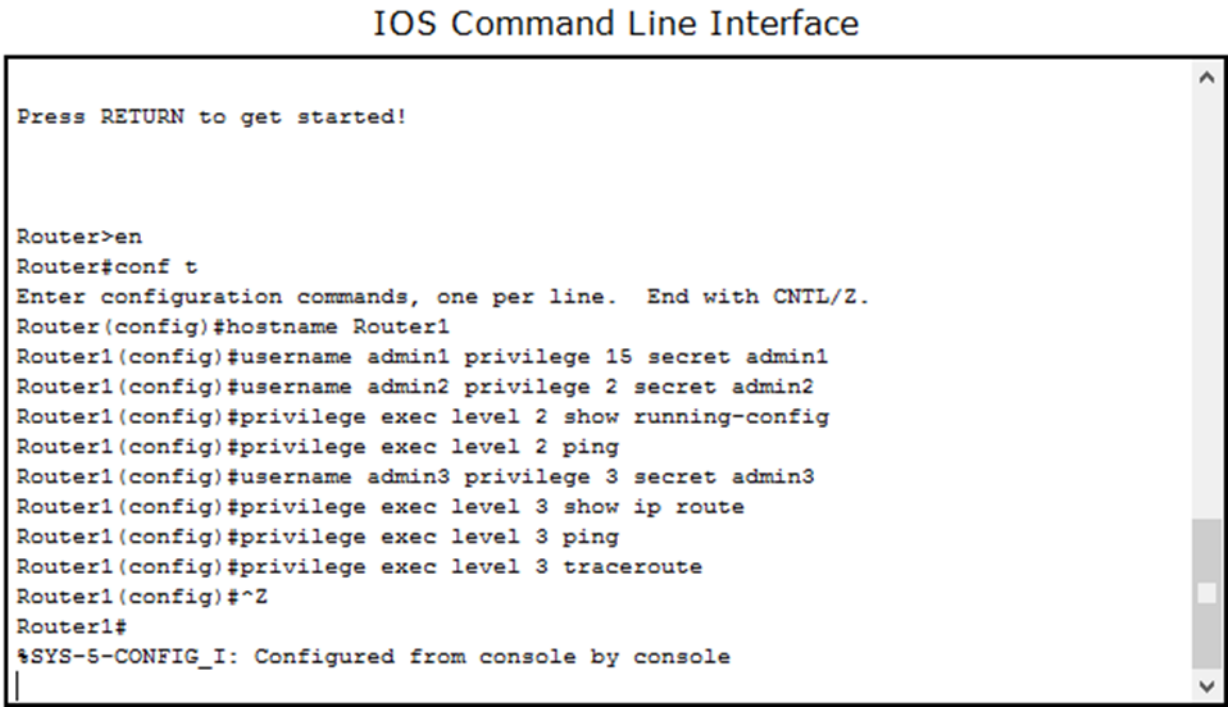


Рисунок 2.6 – Создание локальной базы пользователей

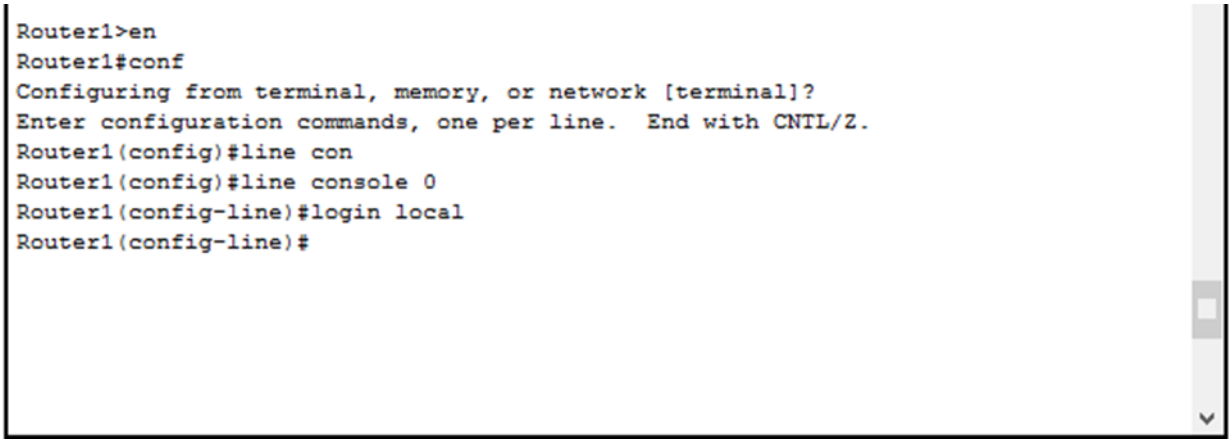


Рисунок 2.7 – Настройка входа по локальной базе

Зайдем на маршрутизатор через консольный порт как admin2 и попробуем выполнить сначала запрещенную, а затем разрешенную для этого пользователя команду, рисунок 2.8.

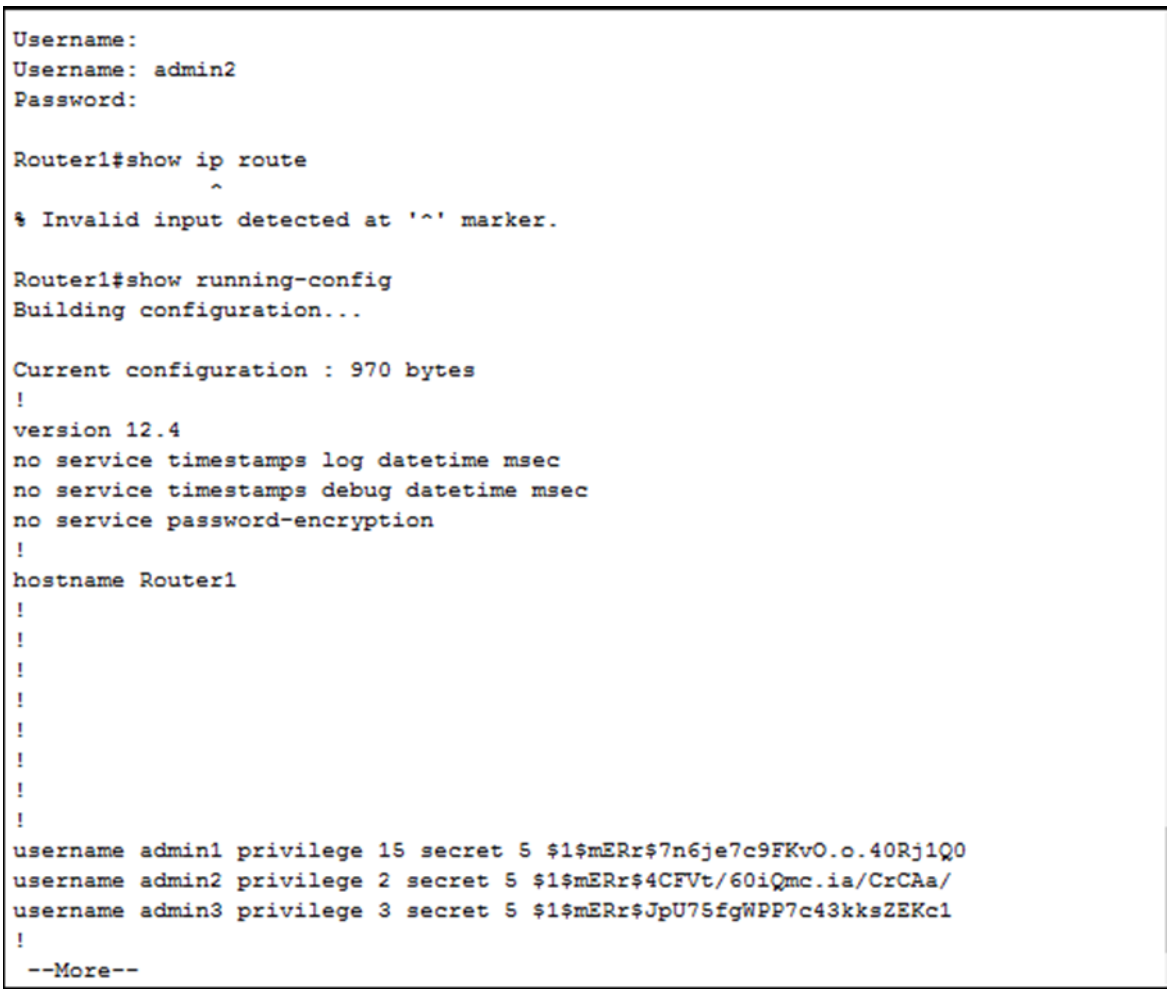


Рисунок 2.8 – Пример выполнения команд

Из рисунка видно, что команда **show ip route** не была выполнена (она запрещена для этого пользователя), а команда **show running-config** вывела информацию о состоянии устройства.

На практике представленный выше способ используется редко. Все современные устройства Cisco поддерживают функцию AAA (Authentication, Authorization and Accounting). Для включения данной функции используется команда

Router1(config)#aaa new-model

Аутентификация настраивается командой

Router1(config)#aaa authentication login <method-list> local

Параметр **local** указывает на необходимость использования локальной базы, параметр **method-list** указывает на используемый метод аутентификации. Например, при использовании метода **default**

аутентификация применяется не только к консольному, но и к удаленному доступу, о котором пойдет речь ниже.

В заключение особенностей конфигурирования консольного доступа следует отметить следующее. Практически все современное оборудование имеет функцию сброса к заводским установкам. При этом могут быть сброшены и установленные пароли. В устройствах Cisco сброс паролей может быть запрещен соответствующей командой (**no service password-recovery)**, однако функция сброса пароля может оказаться полезной. Поэтому наряду с техническими мероприятиями по защите доступа к сетевым устройствам обязательно должны реализовываться и организационные – ограничение физического доступа к устройству, использование выделенных серверных комнат, вандалоустойчивых шкафов, и т.д.

2.2 Протоколы удаленного доступа к сетевому оборудованию

Очевидно, что после начального конфигурирования оборудования производить какие-либо изменения в режиме консольного доступа крайне неудобно. Поэтому существуют протоколы, позволяющие получить удаленный доступ. Здесь рассмотрим самые распространенные из них – Telnet и SSH.

Протокол Telnet (Telecommunication Network Protocol) позволяет выполнить удаленный вход на любое устройство, способное работать в качестве сервера Telnet. В этом случае пользователь получает интерактивный рабочий интерфейс, с которым он может работать так же, если бы он подключился по консоли, рисунок 2.9.

Рисунок 2.9 – Иллюстрация возможностей протокола Telnet

Telnet является клиент-серверным протоколом, устанавливающим TCP-соединение между клиентом и сервером, рисунок 2.10.

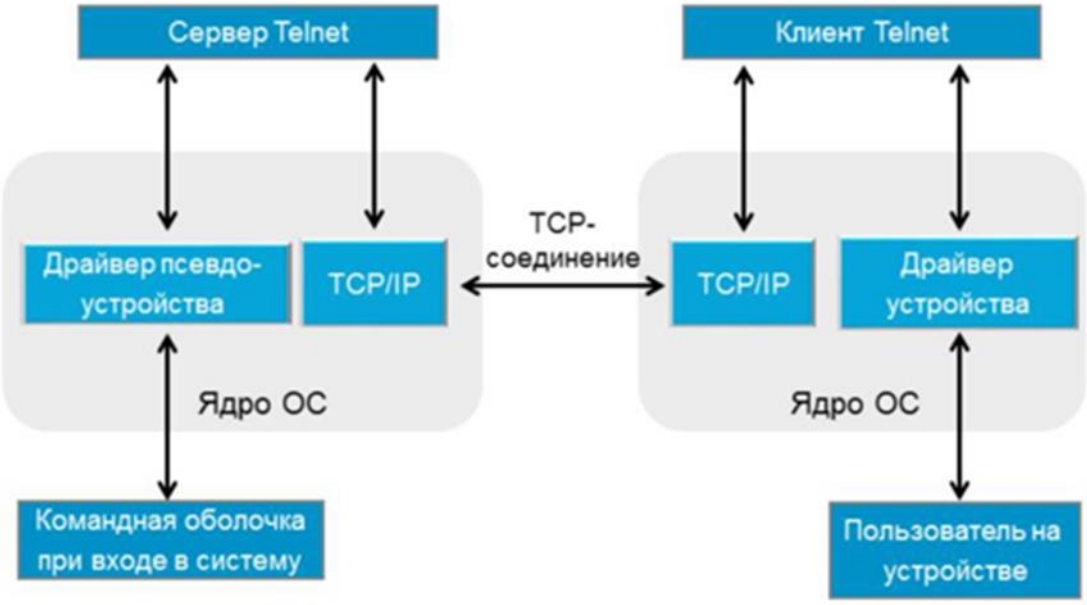


Рисунок 2.10 – Архитектура протокола Telnet

Как известно [3], протокол Telnet является одним из самых старых протоколов удаленного доступа, поэтому сегодня трудно найти оборудование, не поддерживающее этот протокол. Конфигурирование удаленного доступа по протоколу Telnet не является сложным, что является

его несомненным достоинством. Однако данный протокол обладает одним существенным недостатком, существенным с точки зрения безопасности сетей – все данные передаются в открытом виде, что дает возможность злоумышленнику получить доступ как к логинам и паролям, передаваемым в процессе аутентификации, так и непосредственно к передаваемым данным.

С этой точки зрения более предпочтительным является протокол SSH. Протокол SSH передает все данные в зашифрованном виде, что существенно повышает безопасность сети. Предпочтительным является использование последней версии протокола SSHv.2 как наиболее безопасной [4].

Основной задачей протокола SSH в нашем случае является установление удаленного соединения с оборудованием, но, в отличие от Telnet, в защищенном режиме.

Как известно из курса криптографии, для защищенного информационного обмена могут использоваться системы как с симметричным, так и с асимметричным шифрованием. Наиболее криптостойкой является система с симметричным шифрованием, однако главной проблемой таких систем является обеспечение возможности распределения ключей по открытой сети. Асимметричные системы лишены этого недостатка, однако они являются менее криптостойкими. Это означает, что для обеспечения одинаковой криптостойкости в асимметричных системах необходимо использовать более длинный ключ. Соответственно, возрастают и накладные расходы на шифрование и расшифрование передаваемых сообщений.

Протокол SSH является компромиссным решением – при установлении соединения используется асимметричное шифрование, при передаче данных – симметричное.

Рассмотрим это подробнее на примере, рисунок 2.11.

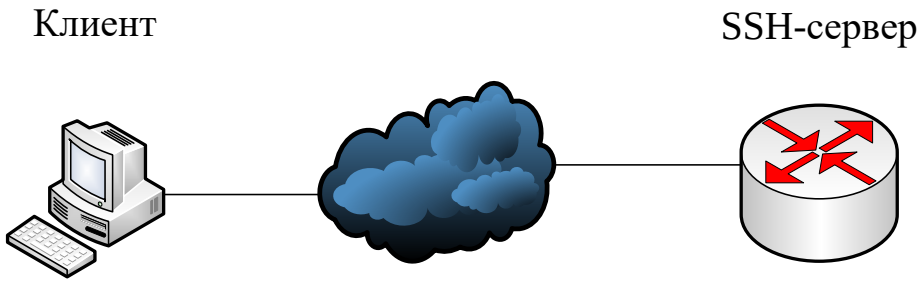


Рисунок 2.11 – Пример использования протокола SSH

В качестве клиента здесь выступает компьютер администратора, в качестве сервера – оборудование (маршрутизатор, коммутатор, и т.д.).

Клиент инициирует единение с сервером, в ответ на это сервер высылает клиенту свой публичный ключ (напомним, в асимметричной системе публичный ключ используется для шифрования, частный – для дешифрования).

Клиент, выбрав по определенному закону, свой ключ для последующего симметричного шифрования, зашифровывает его полученным публичным ключом, и отсылает серверу.

Сервер, получив зашифрованный симметричный ключ, расшифровывает его своим частным ключом. Таким образом, и клиент, и сервер имеют симметричный ключ, используемый для передачи данных.

Симметричный ключ называют сеансовым, так как он действителен только в течение данного сеанса связи. Более того, сеансовый ключ может обновляться с использованием рассмотренной выше процедуры либо через определенный промежуток времени, либо после передачи определенного количества байтов. Диаграмма представлена на рисунке 2.12.

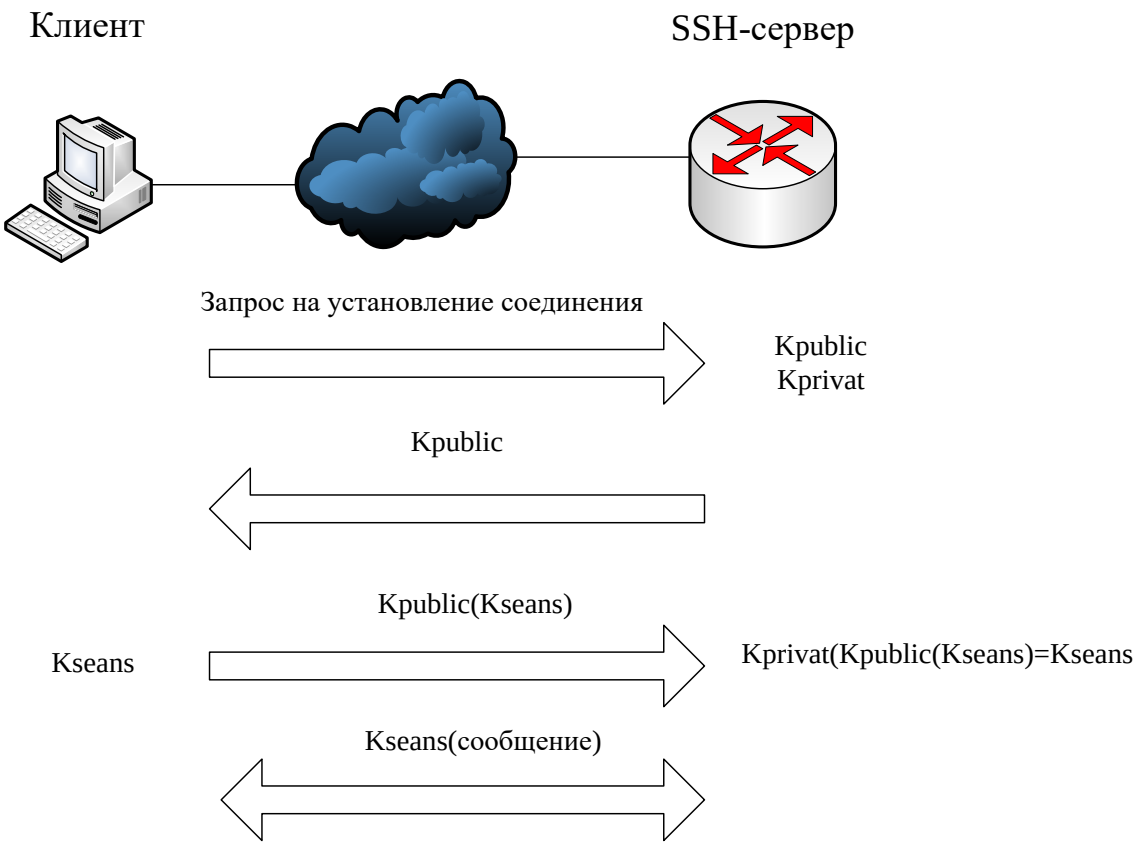


Рисунок 2.12 – Диаграмма обмена по протоколу SSH

На рисунке 2.12 диаграмма представлена в несколько упрощенном виде.

2.3 Защита удаленного доступа

Независимо от используемого протокола удаленного доступа сам доступ организуется с использованием виртуального интерфейса **vty**. Таким образом, необходимо сначала рассмотреть способы конфигурирования виртуальных интерфейсов.

Первый способ аналогичен настройке консольного порта (см. рисунок 1.9) с тем исключением, что обращаться надо не к консольному порту (**line 0**), а к виртуальному (**vty**). Так как к устройству могут удаленно подключаться несколько пользователей, организуется несколько виртуальных интерфейсов (то есть несколько одновременных удаленных подключений). В более старых версиях IOS число виртуальных интерфейсов

5 (от 0 до 4), в более новых – 16 (от 0 до 15), однако для конфигурирования никакой разницы нет. В устройствах Huawei по умолчанию число виртуальных интерфейсов составляет 5, но может быть увеличено до 15 специальной командой.

В этом случае для включения аутентификации по локальной базе пользователей всех пяти виртуальных интерфейсов используются команды:

Router1(config)#line vty 0 4

Router1(config-line)#login local

Второй способ заключается в использовании AAA. Если при конфигурировании консольного порта уже была включена функция AAA с параметром **default**, то аутентификация будет работать как для консольного, так и для виртуальных интерфейсов.

Дальнейшее конфигурирование удаленного доступа зависит от того, какой из протоколов предполагается использовать. Рассмотрим конфигурирование удаленного доступа на базе протоколов Telnet и SSH.

Рассмотрим фрагмент сети, показанный на рисунке 2.13. Для моделирования процесса конфигурирования удаленного доступа будем использовать сетевой эмулятор GNS3 [4], а не Cisco Packet Tracer, ввиду его большей функциональности.

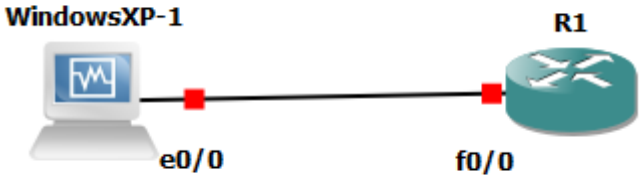


Рисунок 2.13 – Фрагмент сети

Из рисунка 2.13 видно, что рассматриваемый фрагмент сети содержит маршрутизатор (в нашем случае это Cisco 3745), к порту f 0/0 подключена рабочая станция под управлением ОС Windows XP.

Для обеспечения удаленного доступа к маршрутизаторам с использованием этих рабочих станций необходимо на каждом из них создать локальную базу пользователей, и настроить вход через виртуальный интерфейс с аутентификацией по локальной базе. Приведем необходимые для этого команды для маршрутизатора R1 (команды конфигурирования интерфейсов маршрутизатора здесь приводить не будем, считаем, что интерфейсы были сконфигурированы в консольном режиме):

R1(config)#enable secret <пароль> – задаем пароль для входа в привилегированный режим;

R1(config)#username <имя> privilege <уровень> secret <пароль> – создаем пользователя с уровнем привилегий и паролем;

R1(config)#line vty 0 4 – входим в режим конфигурирования виртуальных интерфейсов;

R1(config-line)#login local – задаем режим аутентификации по локальной базе.

Конфигурирование маршрутизатора с использованием консольного порта показано на рисунке 2.14. Как видно из рисунка, для доступа в привилегированный режим был использован пароль **floor1**, в локальной базе создан пользователь **manin1** с уровнем привилегий 1 и паролем **1234**.

Рисунок 2.14 – Конфигурирование маршрутизатора

Рекомендуется использовать именно первый уровень привилегий при входе, так как в этом случае пользователь попадает в непривилегированный режим, и для дальнейшей работы ему необходимо будет ввести еще один пароль (авторизация). Это существенно повышает безопасность доступа к устройству.

После запуска на рабочей станции протокола Telnet было предложено ввести логин и пароль (рисунок 2.15). Так как для пользователя manin1 был определен первый уровень привилегий, данный пользователь после аутентификации был переведен в непривилегированный режим. После набора команды **enable** (переход в привилегированный режим) было предложено ввести дополнительный пароль (рисунок 2.15).

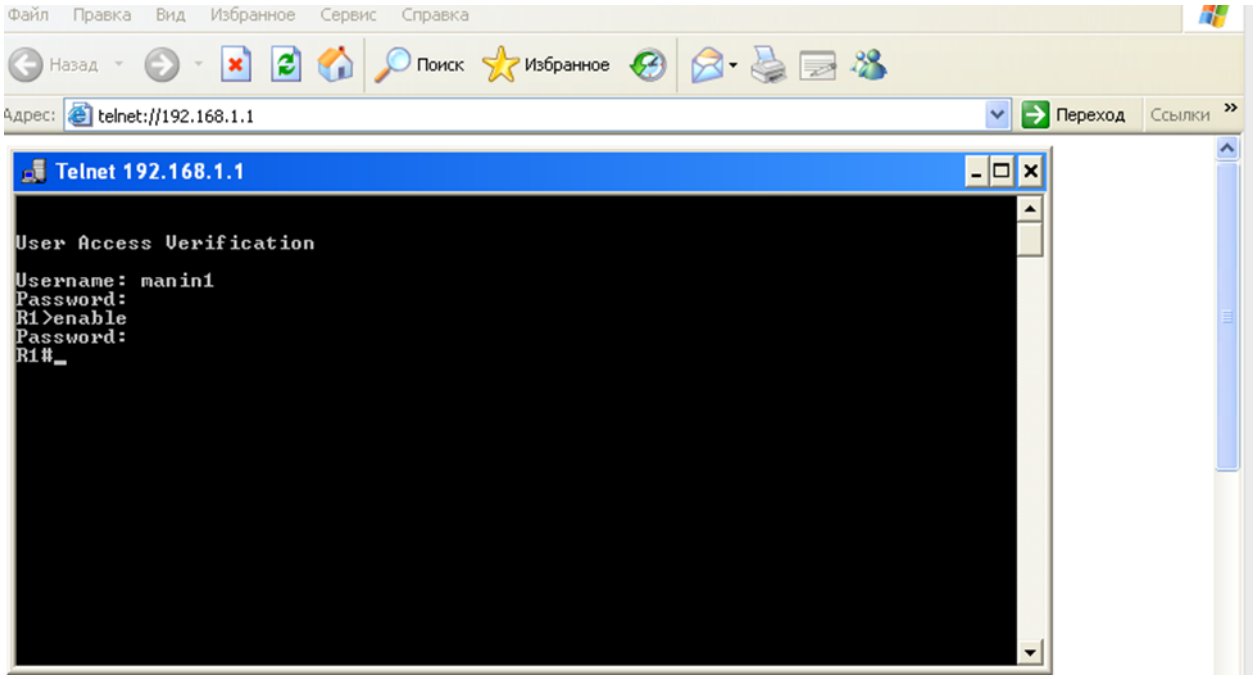


Рисунок 2.15 – Аутентификация пользователя при удаленном доступе

Как указывалось выше, протокол Telnet передает данные в открытом виде, следовательно, их можно перехватить. Для перехвата трафика будем использовать программу WireShark [5]. Результат анализа пакетов, проходящих между рабочей станцией и маршрутизатором при удаленном

доступе с использованием данной программы иллюстрируется рисунком 2.16.

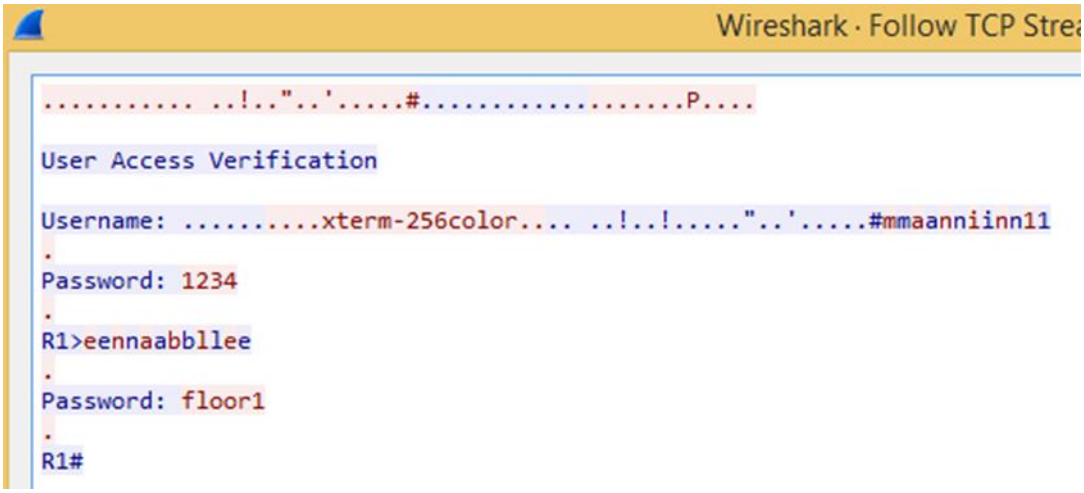


Рисунок 2.16 – Результат анализа передаваемых данных

Из рисунка 2.16 видно, что был использован вход пользователя manin1 с паролем 1234, для входа в привилегированный режим был использован пароль floor1, вход был выполнен успешно. Таким образом, при перехвате сообщений нарушитель получает доступ к логину и паролям. Поэтому протокол Telnet можно использовать только в доверенных сетях.

При использовании SSH маршрутизатор нуждается в дополнительных настройках, так как он должен сгенерировать ключи (см. предыдущий параграф). Для конфигурирования используются следующие команды:

ip domain <имя> - указание имени домена, к которому относится маршрутизатор. Это имя используется для генерации ключей шифрования (совместно с именем маршрутизатора, поэтому при конфигурировании стандартное имя R1 также необходимо изменить).

key generate rsa - генерирует RSA-ключ, после чего IOS просит ввести длину используемого ключа.

transport input ssh - указание маршрутизатору, что для удаленного входа должен использоваться только протокол SSH (в противном случае

конфигурирование SSH теряет смысл, ведь можно будет подключиться и по Telnet).

Пример конфигурирования маршрутизатора показан на рисунке 2.17.

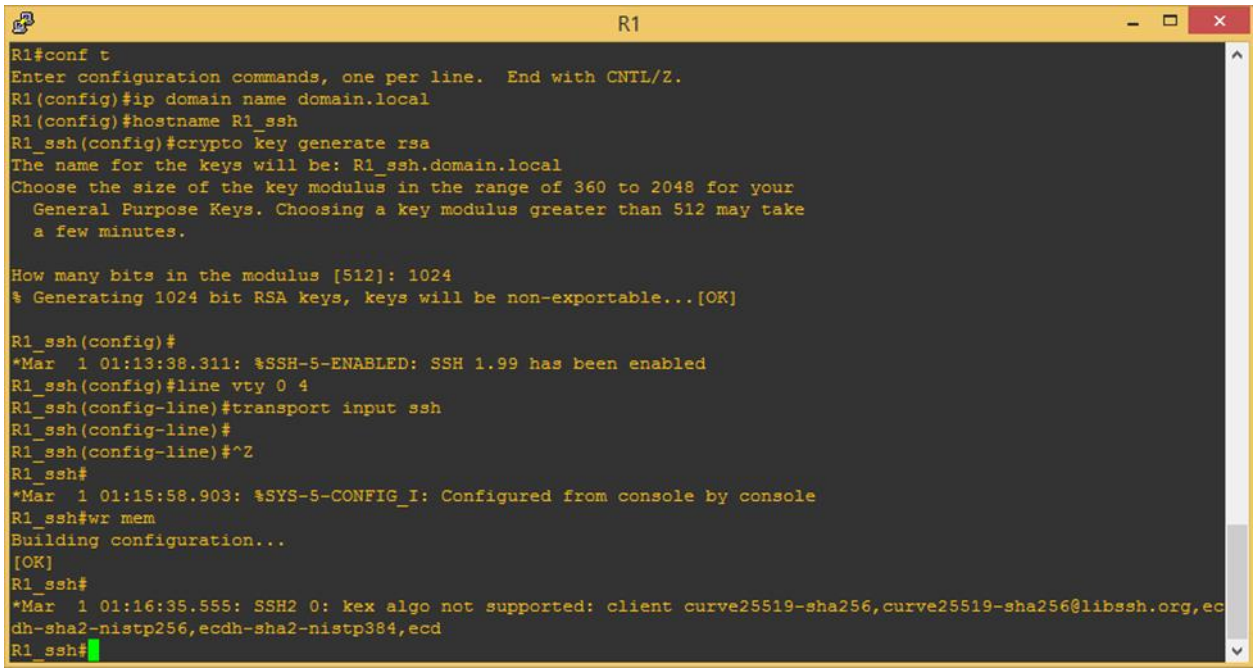


Рисунок 2.17 – Пример конфигурирования маршрутизатора

Удаленный вход на маршрутизатор с использованием программы PuTTY иллюстрируется рисунком 2.18, результат анализа передаваемых данных с использованием программы WireShark – рисунком 2.19.

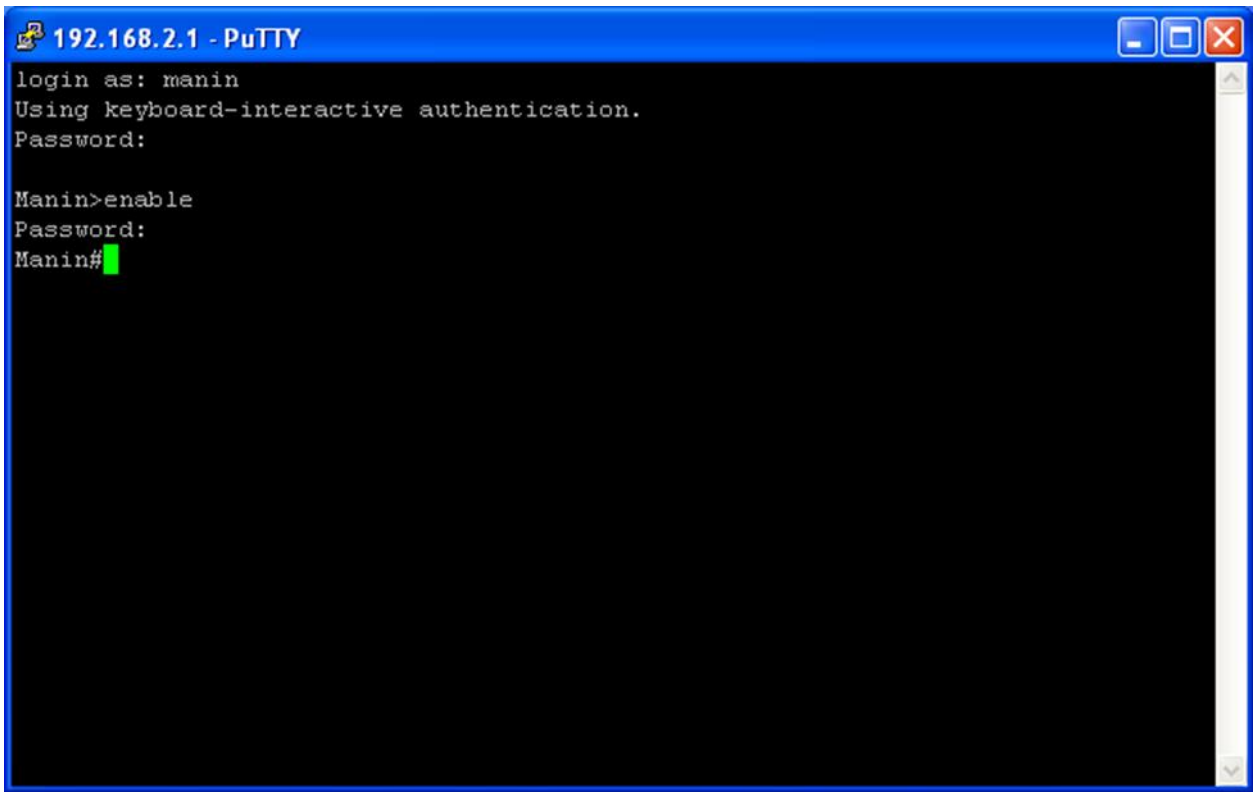


Рисунок 2.18 – Удаленный доступ по протоколу SSH

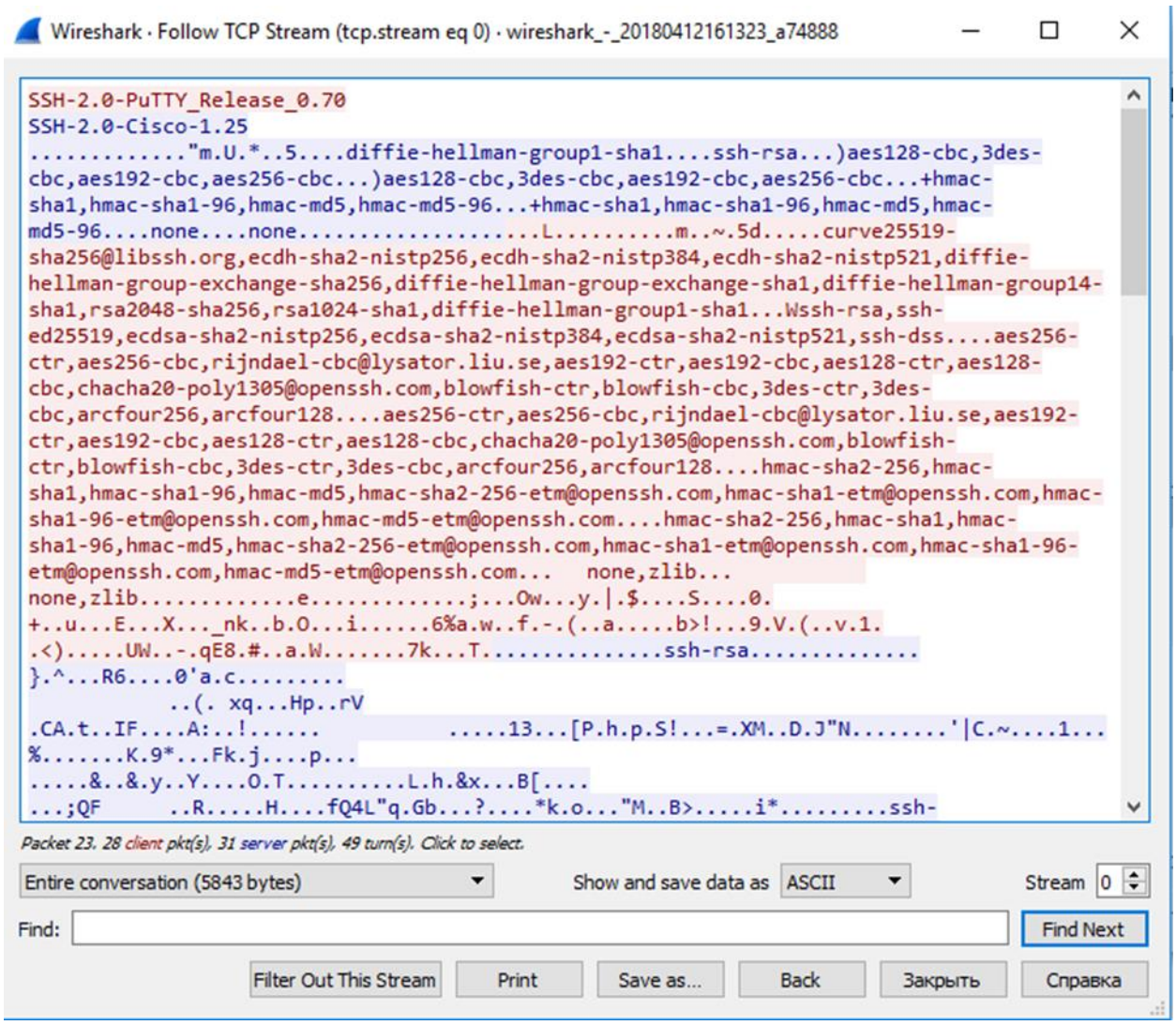


Рисунок 2.19 – Результат анализа передаваемых данных

Из рисунка 2.19 видно, что при использовании протокола SSH перехватить логин и пароли, используемые для удаленного доступа, становится проблематично.

В заключение необходимо отметить следующее. Все современные сетевые устройства позволяют настроить аутентификацию удаленного доступа с использованием более нового и универсального метода, который получил название **aaa new-model**. Основное отличие его состоит в том, что при настройке создается так называемый метод аутентификации (method-list), который определяет, как будет происходить аутентификация при различных способах доступа. Если в качестве method-list используется **default**,

назначенный способ аутентификации будет работать при любых способах доступа к устройству.

В рассмотренном выше примере можно было использовать команды:

R1(config)#aaa new-model

R1(config)#aaa authentication login default local

В этом случае появляется возможность для разных способов доступа (consol, vty, aux) задавать свои методы аутентификации. В нашем примере при использовании любого способа доступа будет применена аутентификация по локальной базе пользователей (параметр **local**).

Кроме того, на практике не стоит пренебрегать дополнительными возможностями ограничения доступа, которые предоставляет IOS. Например, можно ограничить тайм-аут сессии, число и интервал времени, в течение которого можно повторно вводить пароль в случае ошибки, и т.д. Приведем здесь наиболее часто употребляемые команды:

R1(config-line)#exec-timeout 5 0 – величина тайм-аута сессии (5 мин. 0 сек.);

R1(config)#ip ssh time-out 15 – ограничение времени ввода логина и пароля;

R1(config-line)#transport input ssh – разрешение удаленного входа только по протоколу SSH (Telnet работать не будет);

R1(config)#login delay 5 – задание интервала между повторными вводами пароля;

R1(config)#login block-for 60 attempts 3 within 30 – блокировка входа на 60 секунд, если в течение 30 секунд было 3 неудачных попытки входа.

Кроме аутентификации и авторизации, на локальном устройстве может быть настроен и учет (Accounting), хотя чаще он используется с AAA-сервером. Для этого включается так называемое логирование команд, а сами логи локально хранятся на устройстве.

Команды:

R1(config)#archive – включение функции хранения команд;

R1(config-archive)#log config – вход в режим конфигурирования архива;

R1(config-archive-log-cfg)#logging enable – включение функции логирования вводимых команд;

R1(config-archive-log-cfg)#logging size 200 – указание на количество показываемых строк;

R1(config-archive-log-cfg)#hidekeys – указание на «вырезание» паролей.

Просмотр архива выполняется при помощи команды **show archive log config all**.

2.4 Практическое задание

2.4.1 Используя программные продукты (Cisco Packet Tracer или GNS3) или реальное оборудование, подключиться к консольному порту устройства (коммутатора или маршрутизатора). Создать пароль для входа в привилегированный режим, используя параметр **secret**. Создать три учетные записи с разными уровнями привилегий, используя функцию AAA. При создании учетных записей использовать логин **<фамилия в английской транскрипции№учетной записи>**. Проверить работоспособность системы аутентификации по локальной базе пользователей.

2.4.2 Подключить к одному из Etrhernet-портов маршрутизатора рабочую станцию, произведя необходимые настройки (IP-адреса порта маршрутизатора и рабочей станции). Осуществить удаленное подключение к маршрутизатору с использованием протокола Telnet. Проверить работоспособность системы аутентификации.

2.4.3 Сконфигурировать маршрутизатор для работы с протоколом SSH, исключив возможность использования протокола Telnet. Используя SSH-клиент, осуществить удаленное подключение к маршрутизатору с использованием протокола SSH. Проверить работоспособность системы аутентификации.

ПРИМЕЧАНИЕ. Если для выполнения задания 2.4.3 используется Cisco Packet Tracer, для подключения рабочей станции по протоколу SSH необходимо использовать командную строку и команду:

PC> ssh –l <логин> <IP-адрес>

При использовании эмулятора GNS3 или реального оборудования необходимо использовать любой SSH-клиент, например, PuTTY.

3. Использование AAA-сервера для защиты доступа к сетевому оборудованию

3.1 Основные понятия

Как указывалось в предыдущей главе, помимо использования локальной базы пользователей, для аутентификации и авторизации пользователей широкое применение находят AAA-серверы.

Использование AAA-сервера является более удобным, безопасным, а также масштабируемым способом. Во-первых, для добавления, исключения или изменения прав пользователей в этом случае нет необходимости обращаться к каждому из множества сетевых устройств. Во-вторых, база, хранящаяся на сервере, лучше защищена. В-третьих, при периодической смене паролей обращаться приходится только к серверу, а не ко всему множеству сетевых устройств.

Общий алгоритм аутентификации и авторизации пользователей состоит в следующем.

1. Пользователь пытается подключиться к сетевому устройству (например, маршрутизатору) с использованием одного из рассмотренных выше способов (например, с использованием протокола SSH), вводя свои учетные данные (логин, пароль).

2. Маршрутизатор обращается к AAA-серверу и передает ему учетные данные пользователя.

3. AAA-сервер отыскивает в своей базе соответствующую учетную запись (Authentication) и определяет уровень привилегий (Authorization).

4. AAA-сервер пересылает маршрутизатору соответствующую информацию (успешность аутентификации, уровень привилегий).

5. AAA-сервер фиксирует данное событие (Accounting)

6. Маршрутизатор открывает доступ пользователя к оборудованию с учетом уровня привилегий.

Очевидно, что в этом случае информационный обмен, содержащий конфиденциальные данные (логин, пароли) происходит не только между компьютером пользователя и маршрутизатором, но и между маршрутизатором и AAA-сервером. Очевидно, что для этого необходимо использовать протоколы, надежно защищающие передаваемые данные от перехвата. К таким протоколам относятся RADIUS, DIAMETER, TACACS+, и ряд других.

3.2 Использование протоколов RADIUS и TACAS+

Необходимо отметить, что и RADIUS и TACAS+, как и другие, не рассматриваемые здесь протоколы удаленного доступа, разрабатывались не для доступа именно к сетевым устройствам, а вообще к любым сетевым ресурсам. Поэтому общая схема организации удаленного доступа иллюстрируется рисунком 3.1.

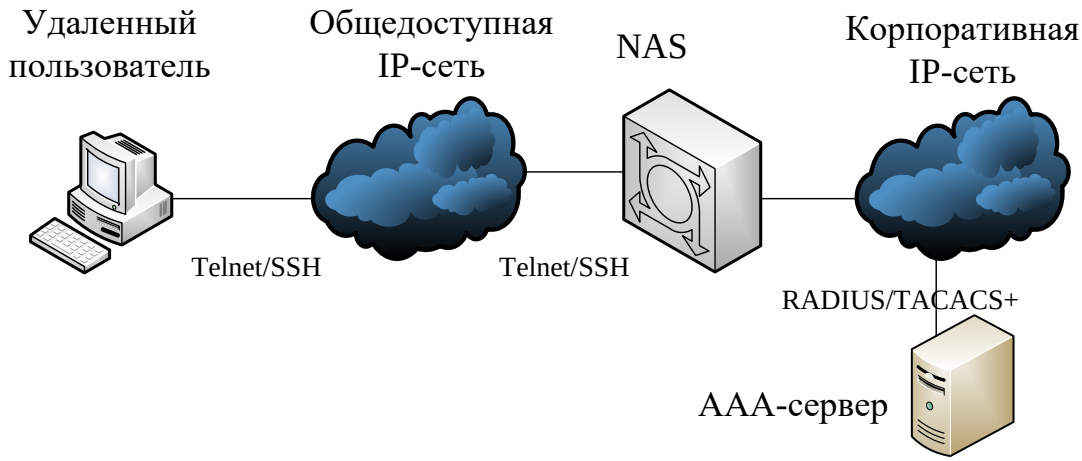


Рисунок 3.1 – Организация удаленного доступа

На схеме показан удаленный пользователь, пытающийся получить доступ к ресурсам корпоративной сети. Для этого он обращается к NAS (Network Access Server – сервер удаленного доступа). NAS в общем случае является сетевым элементом, обеспечивающим удаленный доступ (Remote Access). Информационное взаимодействие между удаленным пользователем

и NAS осуществляется с использованием рассмотренных выше протоколов – Telnet или SSH. При этом взаимодействии удаленный пользователь является клиентом, а NAS – сервером.

Может быть и обратная ситуация, когда пользователь находится во внутренней сети, и запрашивает доступ к внешним ресурсам, например, к Интернет.

При взаимодействии NAS с AAA-сервером NAS является клиентом, а AAA-сервер – сервером. Соответственно, взаимодействие осуществляется с использованием протоколов RADIUS или TACACS+.

В случае доступа именно к маршрутизатору он сам и является NAS, запрашивая доступ к самому себе у AAA-сервера.

Протокол RADIUS (Remote Authentication in Dial-In User Service) передает данные в составе UDP-сегментов (порты 1812 и 1813) и работает по сценарию клиент-сервер. Упрощенная диаграмма информационного обмена показана на рисунке 3.2.

Особенностью протокола RADIUS является то, что процессы аутентификации и авторизации логически не отделены друг от друга. Если пользователь аутентифицирован, ему автоматически предоставляются те права, которые имеются в его учетной записи.

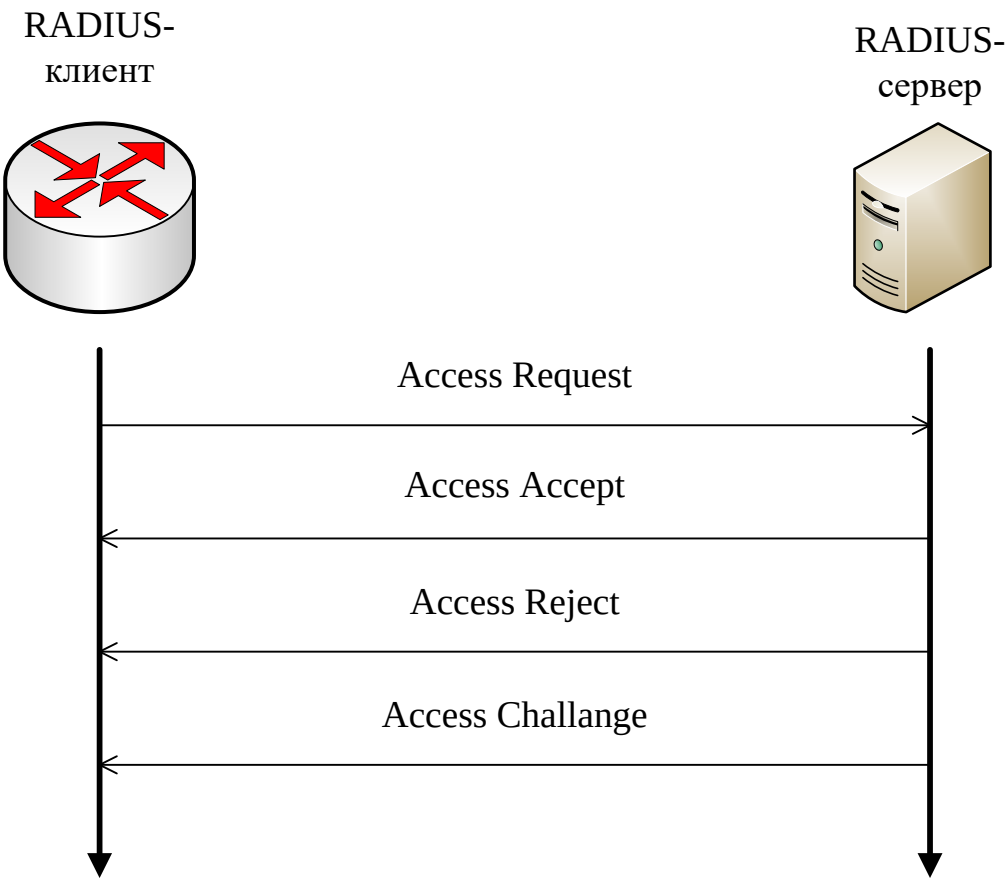


Рисунок 3.2 – Информационный обмен по протоколу RADIUS

Как видно из рисунка, на запрос клиента (Request) сервер может выдать разрешение на доступ (Accept), запрет доступа (Reject), либо запросить дополнительную информацию (Challenge).

Особенностью протокола является то, что при передаче шифруются не все данные, а только пароль. Кроме того, RADIUS является открытым протоколом, поэтому его поддерживает практически все современное оборудование.

Протокол TACACS+ (Terminal Access Controller Access Control System) является разработкой Cisco Systems, обеспечивает передачу данных в составе TCP-сегментов (порт 49) и шифрует все передаваемые данные.

Кроме того, процессы аутентификации и авторизации в этом протоколе логически разделены, что делает его гораздо более гибким, чем

протокол RADIUS. Информационный обмен при аутентификации иллюстрируется рисунком 3.3.

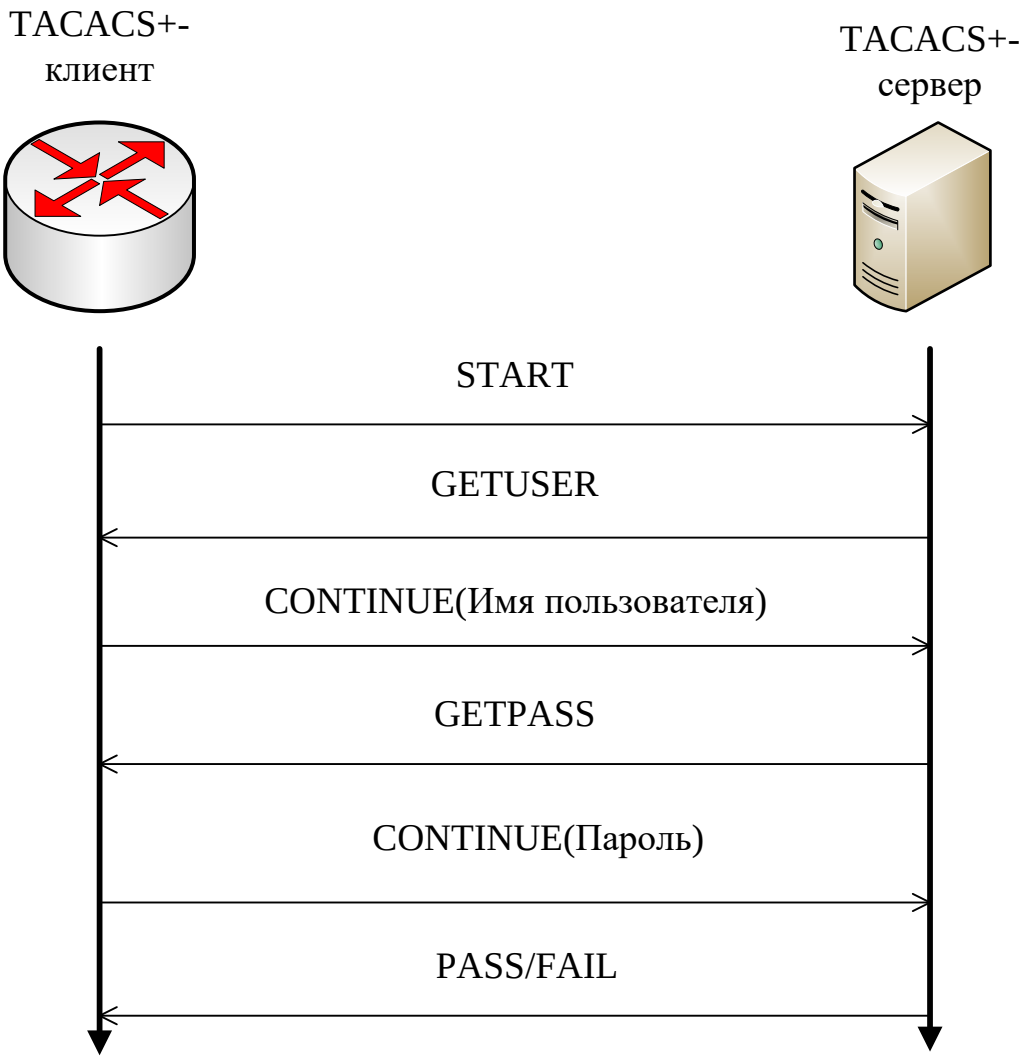


Рисунок 3.3 – Информационный обмен при аутентификации

NAS (в качестве которого в нашем случае выступает маршрутизатор) посылает серверу пакет START для начала процесса аутентификации. Сервер в ответ отсылает пакет GETUSER, содержащий в себе запрос имени пользователя. Маршрутизатор запрашивает имя пользователя у удаленного клиента, получает его (например, с использованием протокола SSH), и отправляет серверу в пакете CONTINUE.

Сервер в пакете GETPASS отправляет запрос на пароль. Маршрутизатор отправляет запрос пароля пользователю, получает его, и отправляет серверу в пакете CONTINUE.

Сервер сверяет полученные имя и пароль со своей базой, и отправляет маршрутизатору либо подтверждение аутентификации (PASS), либо отказ (FAIL).

В процессе авторизации TACACS+ используется два типа пакетов: REQUEST (запрос) и RESPONSE (ответ). Данный процесс авторизации пользователя контролируется посредством обмена парами «атрибут/значение» между сервером защиты TACACS+ и сервером сетевого доступа. Рассмотрим процесс авторизации TACACS+, в котором сервер сетевого доступа обменивается пакетами авторизации с сервером TACACS+, рисунок 3.4.

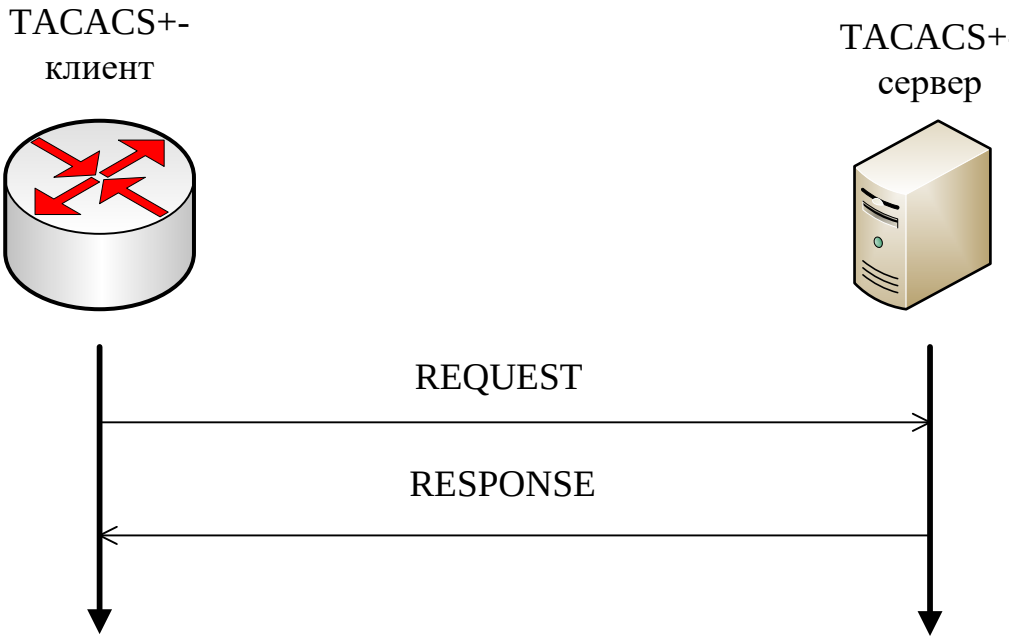


Рисунок 3.4 – Информационный обмен при авторизации

Сервер сетевого доступа (клиент) посылает пакет REQUEST серверу TACACS+. Данный пакет содержит фиксированный набор полей, идентифицирующих пользователя или процесс, а также переменный набор аргументов, описывающих сервисы и параметры, необходимые для авторизации.

Сервер TACACS+ возвращает клиенту пакет RESPONSE, содержащий переменный набор аргументов ответа (пары "атрибут/значение"). Эти пары

строятся на основе ранее заданных разрешений для данного пользователя, хранимых в файле конфигурации TACACS

Сервер сетевого доступа использует пары "атрибут/значение" для того, чтобы запретить, разрешить или модифицировать возможности использования команд и сервисов, запрашиваемых пользователем.

Рассмотрим процесс конфигурирования AAA-сервера, используя бесплатный проект [6]. Для этого в GNS3 соберем схему, показанную на рисунке 3.5.

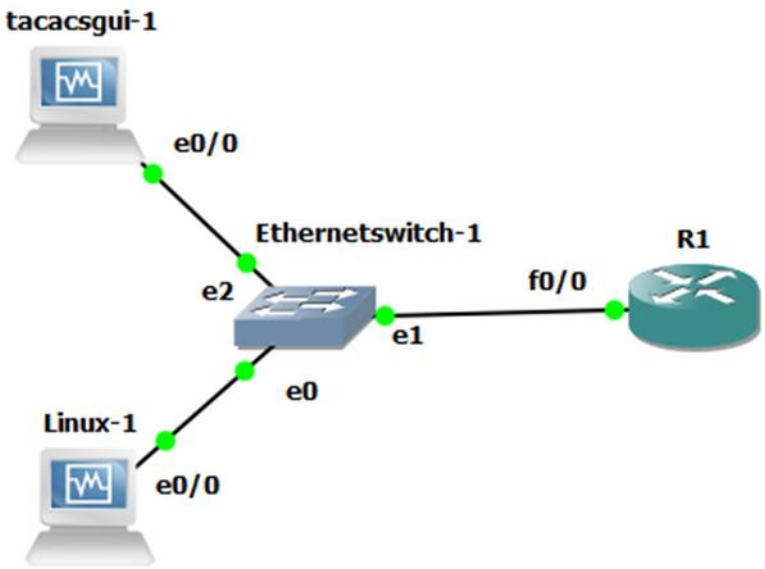


Рисунок 3.5 – Схема сети

Как видно из рисунка 3.5, в схеме имеется AAA-сервер (tacacsgui-1) и рабочая станция (Linux-1), с которой будет осуществляться удаленный доступ к маршрутизатору R1.

Общий вид графического интерфейса tacacsgui показан на рисунке 3.6. Для доступа к нему использовался Интернет-браузер Mozilla Firefox и сокет 10.6.20.10:8008.

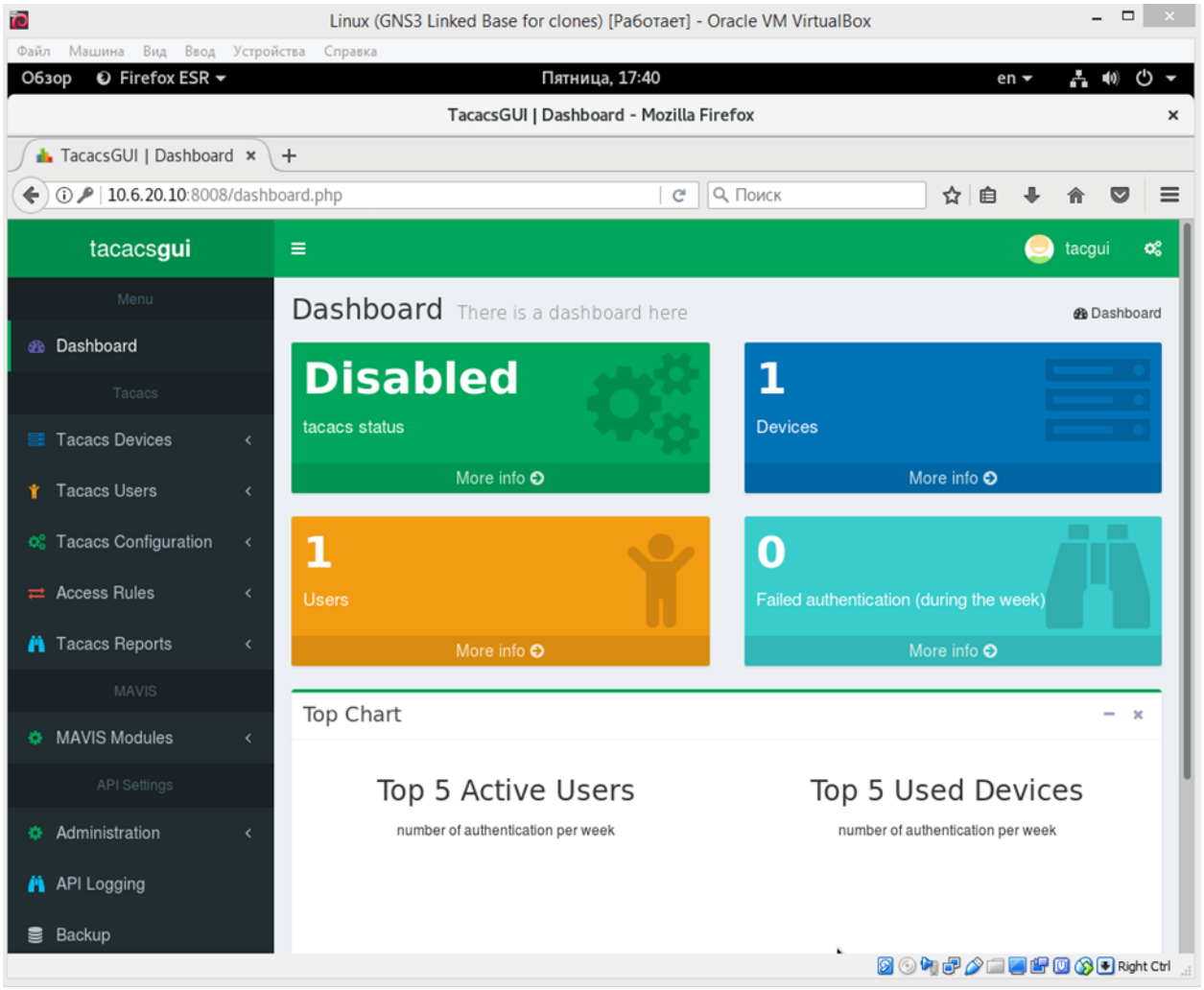


Рисунок 3.6 – Графический интерфейс AAA-сервера

Рисунок 3.7 иллюстрирует процесс добавления устройства, к которому необходимо обеспечить доступ (маршрутизатор R1), а рисунок 3.8 – процесс добавления пользователя (пользователь manin1, пароль 1234, пароль доступа в привилегированный режим – cisco).

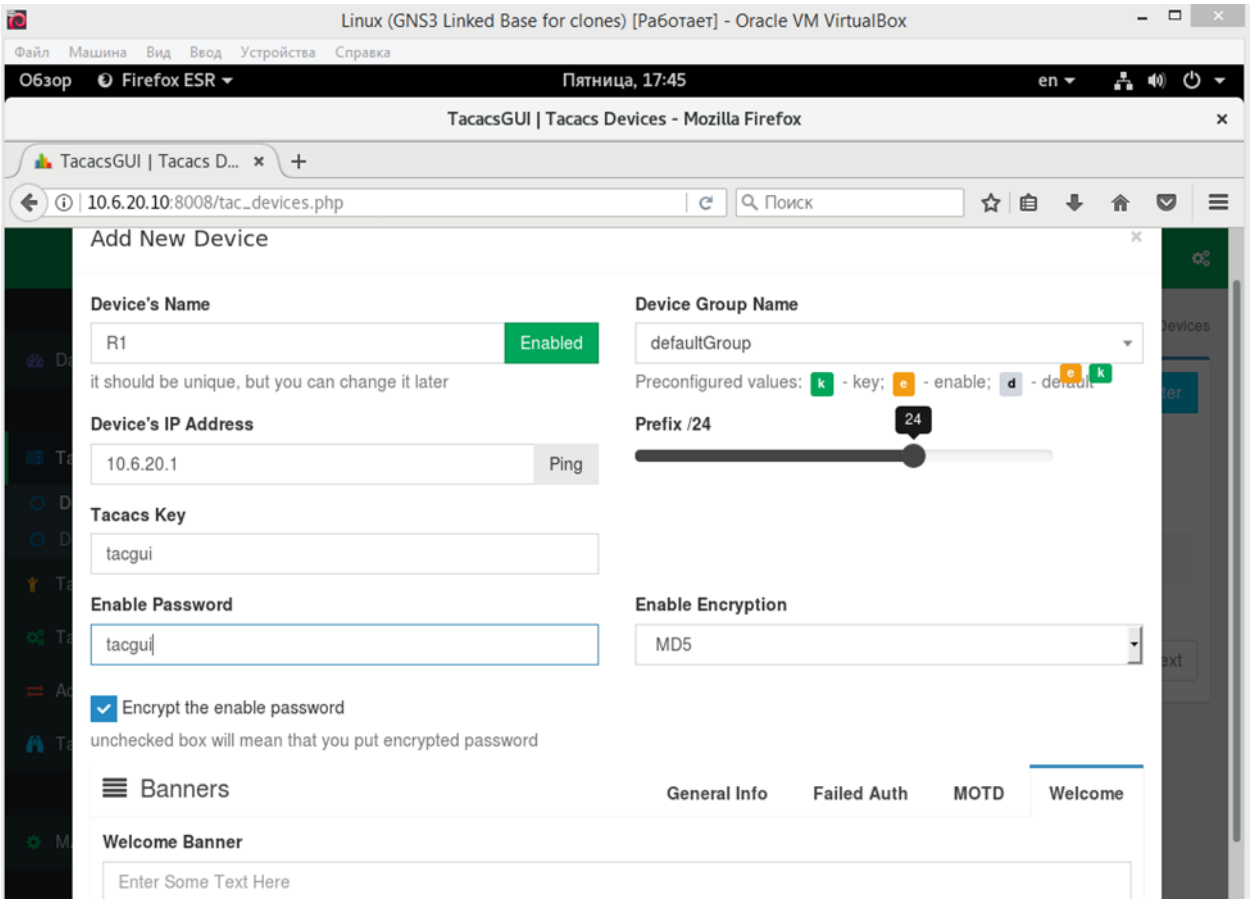


Рисунок 3.7 – Добавление устройства

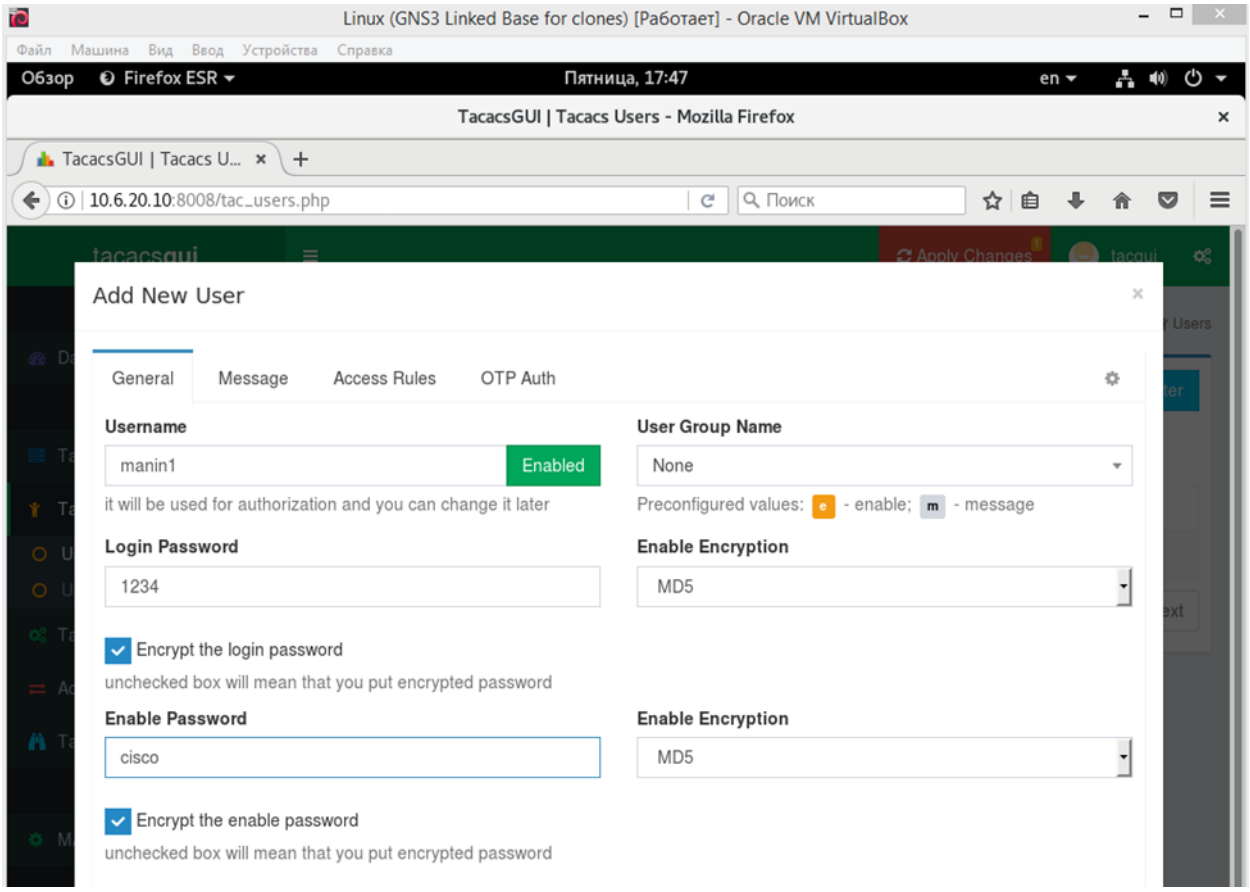


Рисунок 3.8 – Добавление пользователя

Для пользователя manin1 назначим наивысший уровень привилегий 15 (рисунок 3.9).

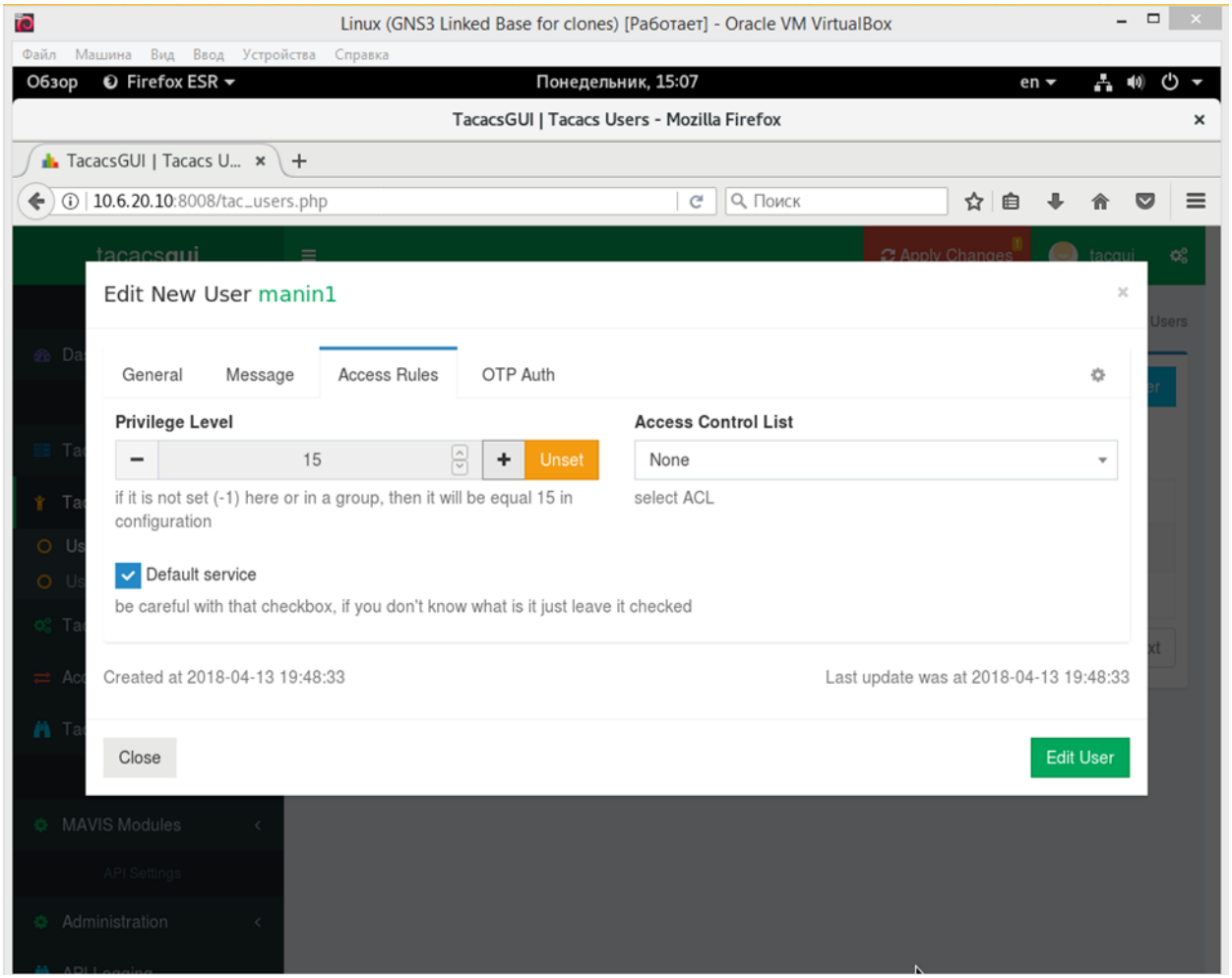


Рисунок 3.9 – Назначение уровня привилегий

На главной вкладке необходимо протестировать введенные конфигурации и применить их, рисунок 3.10.

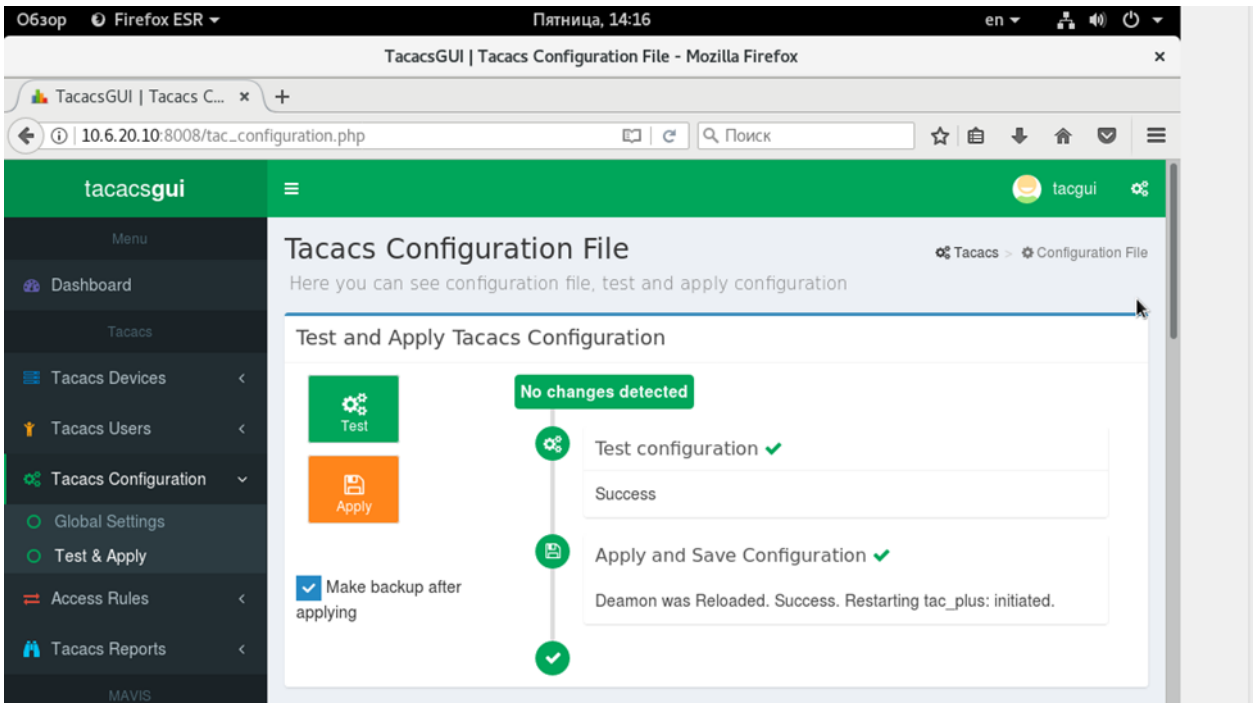


Рисунок 3.10 – Тестирование и применение конфигурации

3.3 Конфигурирование маршрутизатора для работы с протоколом TACACS+

Рассмотрим сначала настройку аутентификации. Для этого необходимо на маршрутизаторе создать локальную учетную запись (на случай, если сервер станет недоступен):

R1(config)#username manin2 privilege 15 secret 1234

Кроме того, зададим пароль для перехода в привилегированный режим:

R1(config)#enable secret cisco

Необходимость создания локальной учетной записи продиктована тем, что при отсутствии технической возможности доступа к AAA-серверу должен быть обеспечен доступ по локальной базе.

Включим режим AAA на маршрутизаторе:

R1(config)#aaa new-model

Затем необходимо указать маршрутизатору параметра используемого AAA-сервера – его IP-адрес и ключ (key), который будет использоваться для шифрования информационного обмена между сервером и маршрутизатором:

R1(config)#tacacs-server host 10.6.20.10

R1(config)#tacacs-server key tacgui

Осталось создать метод аутентификации:

R1(config)#aaa authentication login default group tacacs+ local

Напомним, что параметр **default** в последней команде указывает, что создается method-list по умолчанию, то есть его действие распространяется на все интерфейсы (физические и виртуальные). Параметр **group tacacs+ local** указывает, что аутентификация производится сначала с использованием протокола TACACS+, а при невозможности (например, при недоступности сервера) – с использованием локальной базы.

Конфигурирование маршрутизатора иллюстрируется рисунком 3.11.

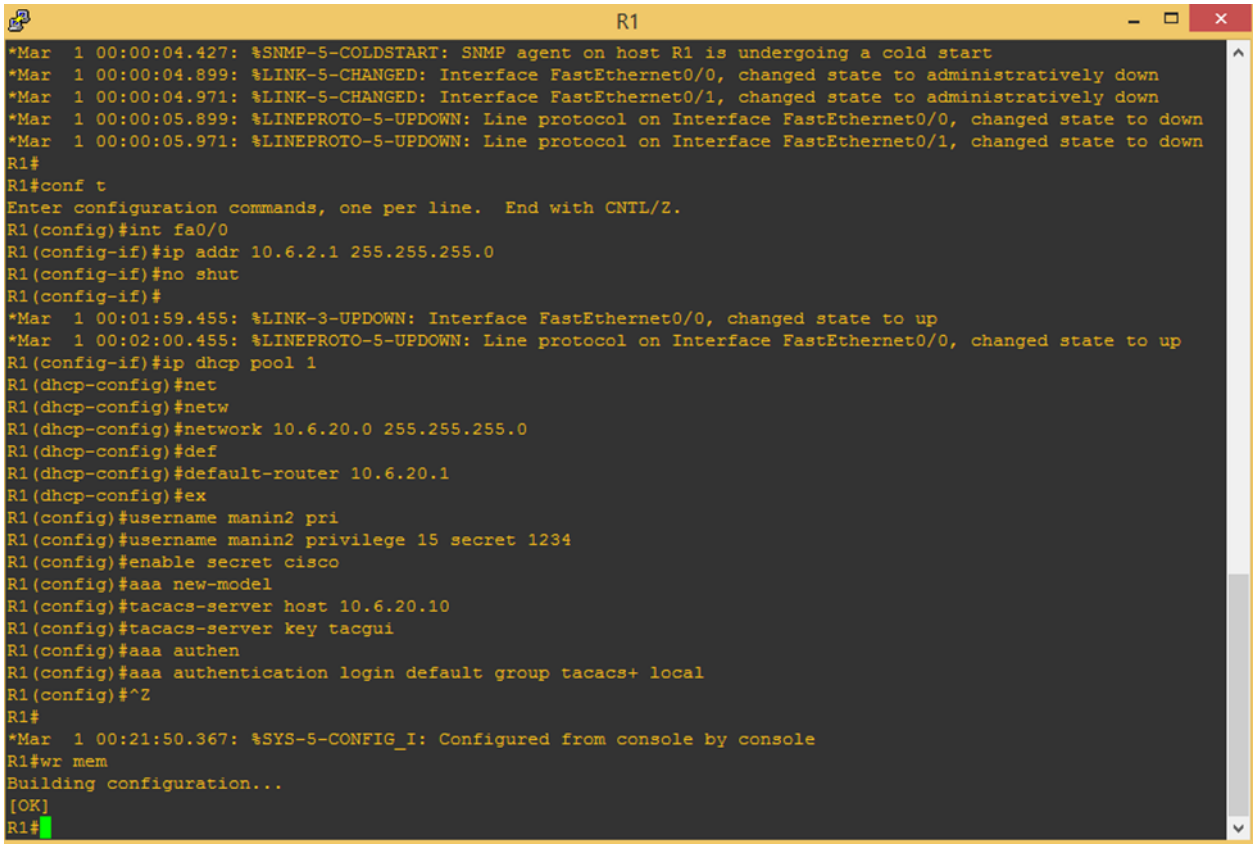


Рисунок 3.11 – Конфигурирование маршрутизатора

Добавим в сеть еще одну рабочую станцию под управлением ОС Windows XP, и попробуем удаленно подключиться к маршрутизатору, используя учетную запись manin1, сконфигурированную на AAA-сервере, рисунок 3.12.

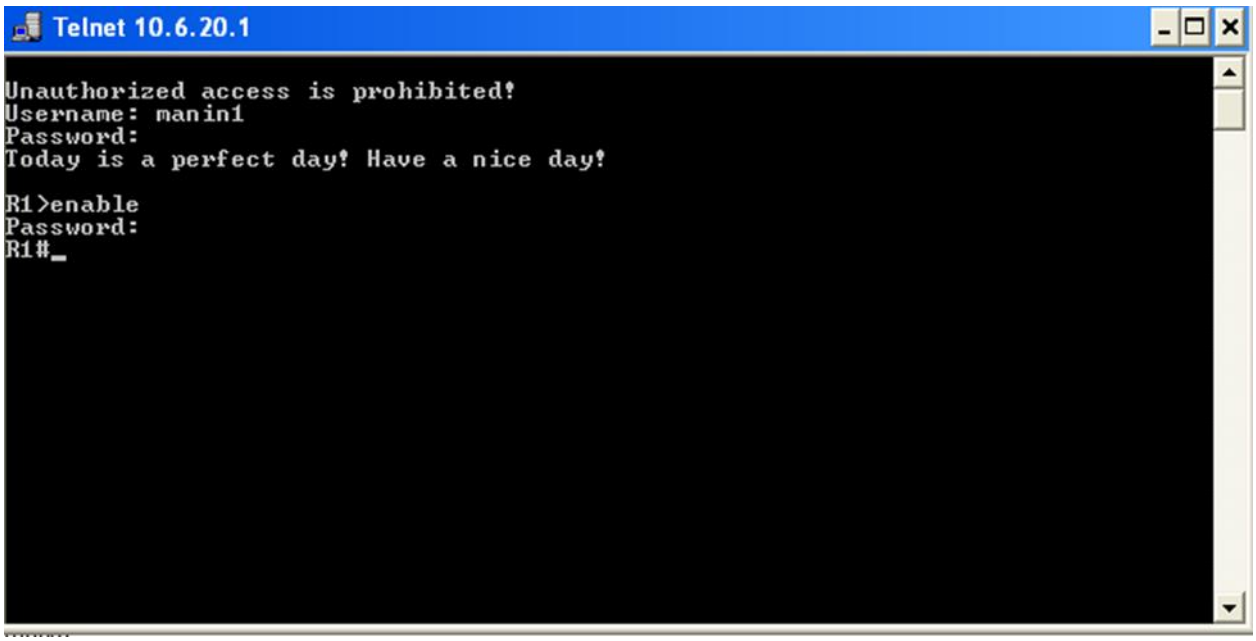


Рисунок 3.12 – Удаленное подключение к маршрутизатору с использованием ААА-сервера

Из рисунка 3.12 видно, что аутентификация прошла успешно. Отключим ААА-сервер и снова попробуем подключиться к маршрутизатору, рисунок 3.13.

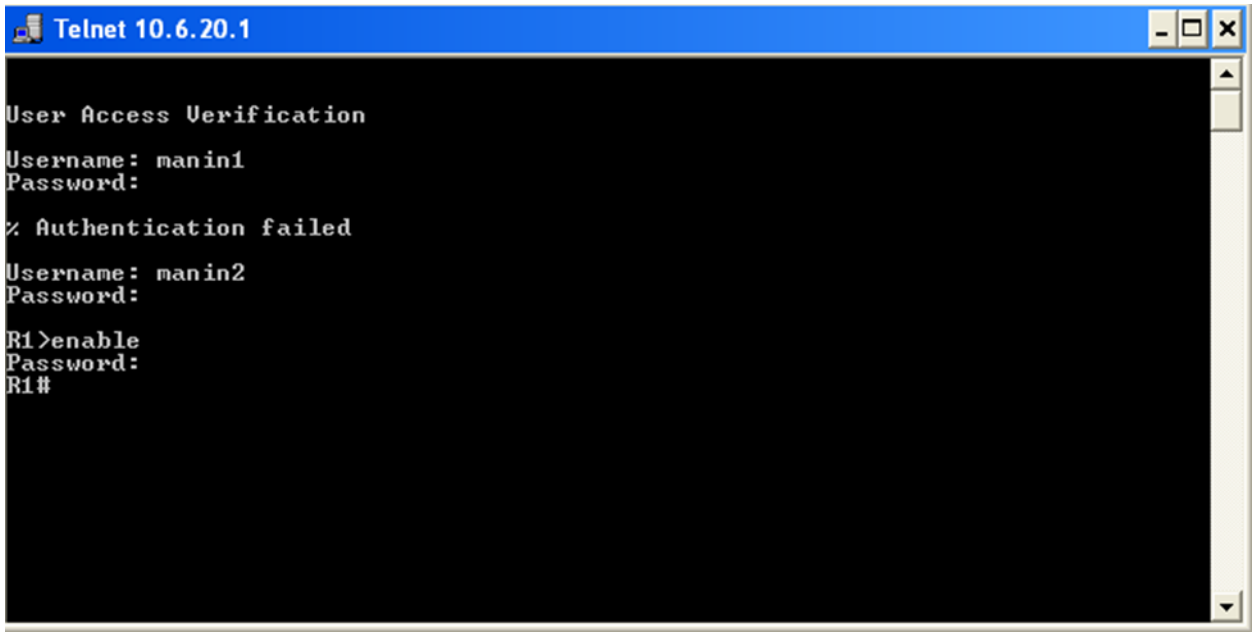


Рисунок 3.13 – Удаленное подключение к маршрутизатору при отключенном ААА-сервере

Как видно из рисунка 3.13, аутентификация по учетной записи manin1, хранящейся на сервере, оказалась безуспешной. Аутентификация по учетной записи manin2, хранящейся в локальной базе, прошла успешно благодаря параметру **group tacacs+ local**, использованному при конфигурировании маршрутизатора.

Протокол TACACS+, в отличие от RADIUS, разделяет процедуры аутентификации и авторизации. Поэтому если для нашего примера дополнительно настроить авторизацию для пользователя manin1, при удаленном доступе этот пользователь сразу попадет в привилегированный режим (так как мы настроили для него 15-й уровень привилегий, рисунок 3.9). Для большей наглядности создадим на AAA-сервере еще одного пользователя manin3 с уровнем привилегий 1 и разрешим ему использование команд **ping, show running-config** и **exit**, рисунок 3.14.

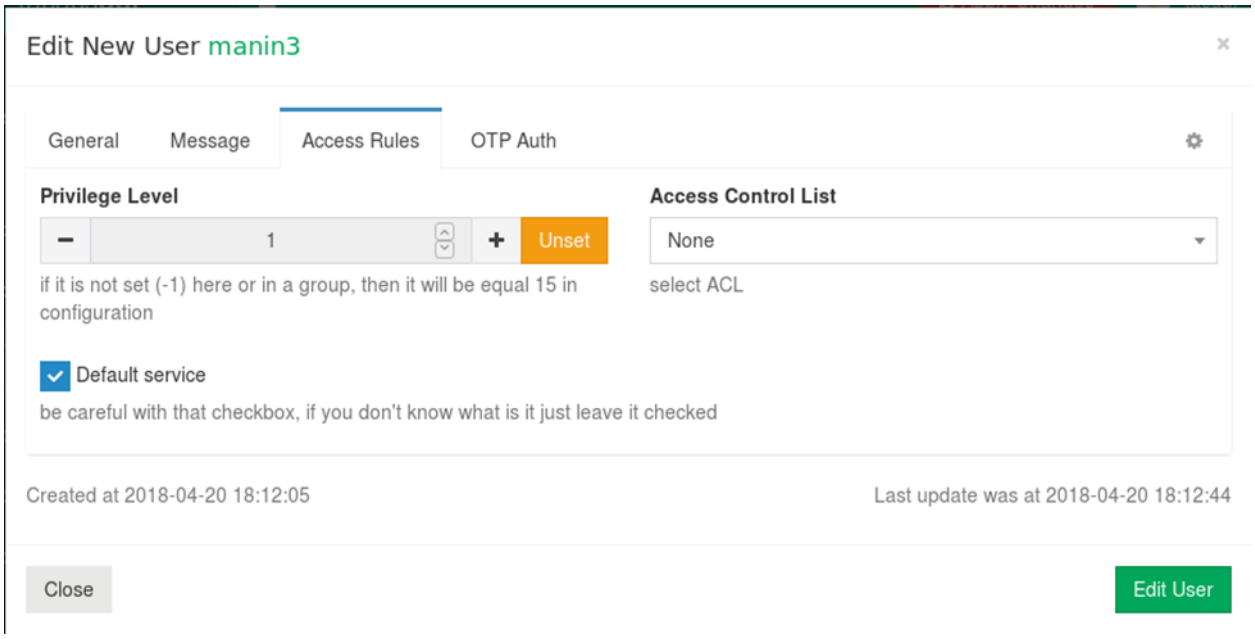


Рисунок 3.14 – Добавление пользователя manin3 с уровнем привилегий 1

Авторизация настраивается с использованием следующих команд:

R1(config)#aaa authorization exec default group tacacs+ local – включение режима авторизации (автоматический переход на нужный режим при подключении к устройству);

R1(config)#aaa authorization config-command – авторизация команд конфигурирования.

R1(config)# aaa authorization commands 1 default group tacacs+ local

R1(config)# aaa authorization commands 15 default group tacacs+ local – авторизация каждой вводимой команды как на уровне 1, так и на уровне 15.

При удаленном подключении пользователя manin1 (уровень 15) мы сразу попадаем в привилегированный режим, рисунок 3.15.

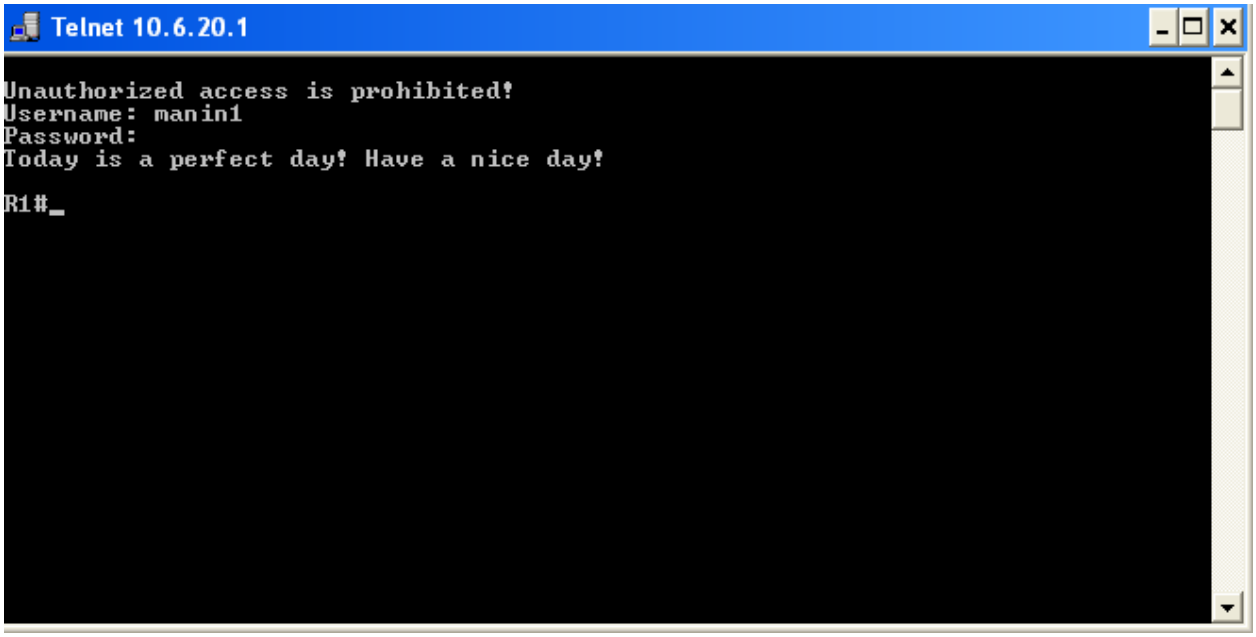


Рисунок 3.15 – Авторизованный вход

Наконец, после авторизации необходимо настроить учет (Accounting). Команды:

R1(config)#aaa accounting exec default start-stop group tacacs+ - включение учета начала/конца каждой сессии;

R1(config)#aaa accounting commands 1 default start-stop group tacacs+ - включение учета команд на уровне 1;

R1(config)#aaa accounting commands 15 default start-stop group tacacs+ - включение учета команд на уровне 15.

Пример просмотра отчета на AAA-сервере показан на рисунке 3.16.

Date/Time	Name Device (IP)	User Name	User IP	Privilege Level	Command
2016-11-30 02:20:23	R1 (192.168.56.254)	admin1	console	1	ping 192.168.56.101
2016-11-30 02:19:24	R1 (192.168.56.254)	admin15	console	15	show running-config
2016-11-30 02:19:24	R1 (192.168.56.254)	admin15	console	15	show running-config
2016-11-30 02:19:17	R1 (192.168.56.254)	admin15	console	15	write memory

Рисунок 3.16 – Просмотр отчета на AAA-сервере

3.4 Логирование сетевых событий

Syslog (System log – системный журнал) – стандарт отправки и регистрации сообщений о происходящих в системе событиях, использующийся в компьютерных сетях, работающих по протоколу IP. Термином «Syslog» называют как стандартизированный сетевой протокол Syslog, так и программное обеспечение (приложение, библиотека), которое занимается отправкой/получением системных сообщений.

Логирование сетевых событий не относится напрямую к вопросам обеспечения безопасности сетей, а чаще применяется для поиска и устранения проблем, возникших в сети (Troubleshooting). Однако с использованием логирования можно также отследить действия потенциальных нарушителей. Например, из просмотра логов можно узнать, сколько было неудачных попыток доступа к сетевым устройствам, кто и когда производил конфигурирование устройств, и т.д.

В случае, если используется оборудование Cisco, могут использоваться следующие способы сбора логов с оборудования:

1. Console Logging – вывод сообщений в консоль.
2. Buffered Logging – сохранение логов в памяти устройства.
3. Terminal Logging – вывод сообщений в терминал в ходе Telnet/SSH сессии.
4. Syslog Server – вывод сообщений серверу Syslog.
5. SNMP Logging – вывод сообщений по протоколу SNMP.

6. AAA – учет с использованием AAA-сервера (способ рассмотрен в предыдущем параграфе).

Кроме того, вводится понятие уровня логирования, который определяет, какая именно информация должна отображаться в логах. Существует восемь уровней логирования:

0 – Emergencies. События, связанные с неработоспособностью системы.

1 – Alerts. Сообщения о необходимости немедленного вмешательства.

2 – Critical. Критические события.

3 – Errors. Сообщения об ошибках.

4 – Warnings. Сообщения с предупреждениями.

5 – Notifications. Уведомления.

6 – Informational. Информационные сообщения.

7 – Debugging. Отладочные сообщения.

Указанные уровни обладают наследственностью. Это означает, что если настроить передачу сообщений с уровнем 4, то будут передаваться все сообщения уровней 0 – 4.

Рассмотрим сначала команды, необходимые для настройки логирования на маршрутизаторе.

Console Logging:

R1(config)# logging console 7

или

R1(config)# logging console debugging

Buffered Logging:

R1(config)#logging on – включение Buffered Logging;

R1(config)#logging buffered 32768 – указание размера буфера для хранения логов;

R1(config)#logging buffered informational – выбор уровня логирования.

Лог-сообщения, хранящиеся в памяти устройства, можно просмотреть командой **show log**, выполняемой из привилегированного режима.

Terminal Logging:

R1(config)#logging monitor informational – указание уровня логирования;

R1(config)#exit

R1#terminal monitor – запуск на маршрутизаторе программы вывода логов терминал.

Syslog Server:

R1(config)#logging host <адрес> - указание адреса Syslog-сервера;

R1(config)#logging trap debugging – указание уровня логирования.

Следует отметить, что само логирование (являющееся в том числе средством защиты) тоже нужно защищать. Как минимум, необходимо обеспечить передачу логов по выделенной VLAN для исключения возможности перехвата (как и всего управляющего трафика). Доступ к Syslog-серверу также должен быть максимально ограничен. Кроме того, сервер желательно защитить от DoS-атак, ограничив число передаваемых логов в единицу времени. Команда:

R1(config)#logging rate-limit all <N> – ограничение до N сообщений в секунду.

В конце данного параграфа необходимо отметить, что логи становятся совершенно бесполезными, если в составе сообщений отсутствует время события, или оно некорректно. Поэтому необходимо, чтобы время на всех устройствах сети было синхронизировано. Наилучшим решением данной задачи является использование протокола Network Time Protocol (NTP). Если в сети есть NTP-сервер, можно указать его в настройках маршрутизатора. Команды:

R1(config)#ntp server <адрес>

R1(config)#clock timezone MSK 4 – указание часового пояса (UTC+4).

Для того, чтобы сами логи были снабжены временными метками, используется команда:

R1(config)#service timestamps log datetime msec localtime show-timezone.

3.5 Практическое задание

3.5.1 Используя Cisco Packet Tracer или GNS3, собрать схему фрагмента сети, показанную на рисунке 3.17. Настроить IP-адресацию, обеспечив доступность сервера с управляющего ПК.

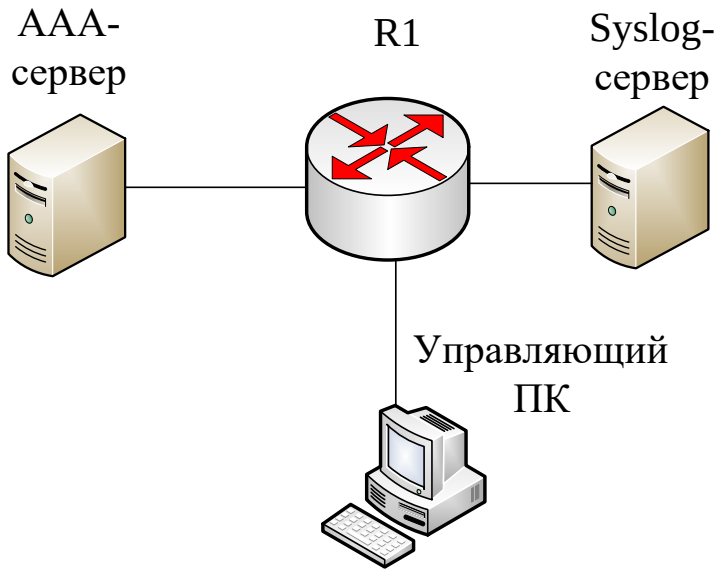


Рисунок 3.17 – Схема фрагмента сети

3.5.2 Сконфигурировать маршрутизатор для работы совместно с AAA- и Syslog-серверами.

3.5.3 Настроить аутентификацию и авторизацию двух пользователей с различными правами, используя протокол TACACS+.

3.5.4 Убедиться в работоспособности системы аутентификации и логирования событий входа-выхода.

4 Межсетевое экранирование

4.1 Определение и классификация межсетевых экранов

Определений межсетевых экранов существует великое множество. Поэтому будем использовать определение, данное в «Руководящем документе. Межсетевые экраны» Гостехкомиссии при Президенте РФ [7].

Межсетевым экраном называется локальное (однокомпонентное) или функционально-распределенное средство (комплекс), которое реализует контроль за информацией, поступающей в автоматизированную систему и/или выходящей из нее, и обеспечивает защиту автоматизированной системы посредством фильтрации информации, т. е. анализа по совокупности критериев и принятия решения об ее распространении в (из) автоматизированной системе.

В обиходе межсетевыми экранами (МСЭ) называют средства защиты, устанавливаемые между общедоступной (такой, как Интернет) и внутренней сетью. Межсетевой экран выполняет двойную функцию.

Во-первых, он призван ограничить доступ во внутреннюю сеть со стороны общедоступной сети за счет применения фильтров и средств аутентификации.

Во-вторых, МСЭ служит для контроля и регулирования доступа пользователей внутренней сети к ресурсам общедоступной сети. Кроме того, МСЭ могут устанавливаться и во внутренней сети для ограничения доступа внутренних пользователей к каким-либо критическим ресурсам. Также МСЭ имеются во всех операционных системах персональных компьютеров (персональные МСЭ).

В настоящее время классификация МСЭ очень разнообразна. Наиболее часто МСЭ классифицируют по уровням модели OSI, на которых они работают. По этому признаку можно выделить следующие типы МАЭ:

1. Управляемые коммутаторы (канальный уровень).

2. Сетевые фильтры (сетевой уровень).

3. Шлюзы сеансового уровня.

4. Посредники прикладного уровня.

5. Инспекторы состояния (сеансовый уровень с расширенными возможностями).

6. МСЭ экспертного уровня (охватывает несколько уровней OSI).

Управляемые коммутаторы можно отнести к МСЭ очень условно. Коммутатор позволяет использовать операцию Security Port, которая заключается в привязке порта коммутатора к определенному MAC-адресу. Это очень ненадежная защита, так как MAC-адрес всегда можно подменить, кроме того, при включении данной опции сеть становится менее масштабируемой, так как отключается функция автоматического обучения коммутатора. Кроме того, коммутатор способен разделить сеть на несколько виртуальных сетей (VLAN), что является более надежным решением, однако оно ограничено пределами одной подсети. Тем не менее, такой способ широко используется в современных сетях, и более подробно будет рассмотрен в следующей главе.

В самом простом случае МСЭ устанавливается на границе между внутренней и общедоступной сетями, рисунок 4.1.

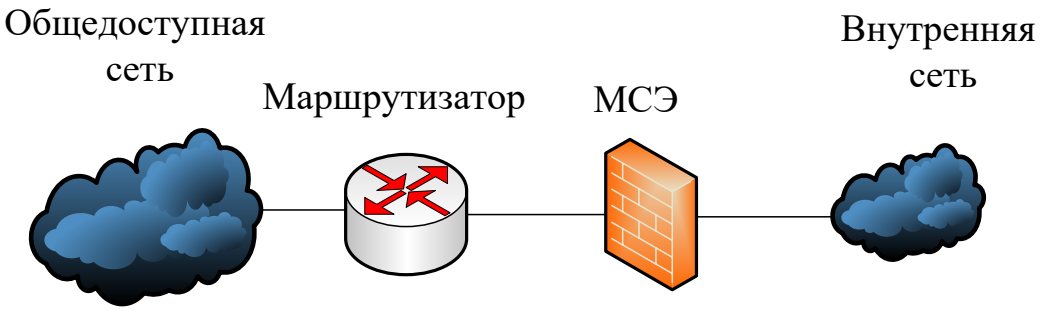


Рисунок 4.1 – МСЭ на границе между внутренней и общедоступной сетями

Такая схема применяется, если для всех узлов внутренней сети требования к защите примерно одинаковы. Например, все соединения из общедоступной сети во внутреннюю запрещены. Если же требования

различны (например, во внутренней сети располагается почтовый или веб-сервер), такая схема не является безопасной.

В этом случае серверы могут быть размещены в открытом сегменте, а остальные узлы – во внутренней сети, рисунок 4.2.

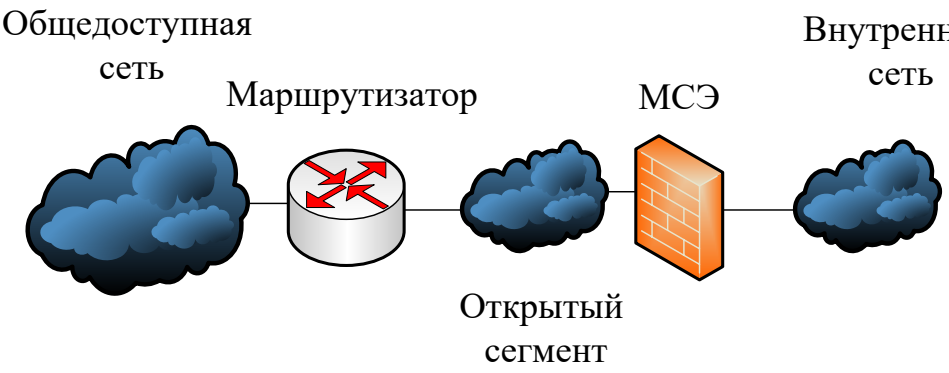


Рисунок 4.2 – МСЭ между открытым и закрытым сегментами внутренней сети

Очевидный недостаток схемы, показанной на рисунке 4.2 – устройства, размещенные в открытом сегменте, не защищены. Выходом здесь может служить применение двух МСЭ, один из которых защищал бы открытый сегмент, а другой – закрытый. Тогда схема подключения МСЭ приобретает вид, показанный на рисунке 4.3.

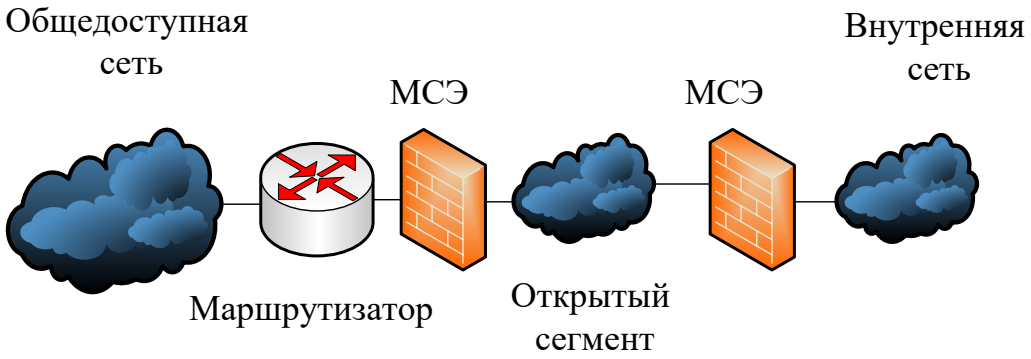


Рисунок 4.3 – Схема с двумя МСЭ

Схема, показанная на рисунке 4.3, является наиболее гибкой в настройке, так как можно задать правила межсетевого экранирования во всех направлениях. В связи с этим многие производители начали выпуск

межсетевого экрана с интерфейсом, специально выделенным для подключения открытого сегмента, а сам такой сегмент получил название «демилитаризованная зона» (DMZ). Схема подключения такого МСЭ показана на рисунке 4.4.

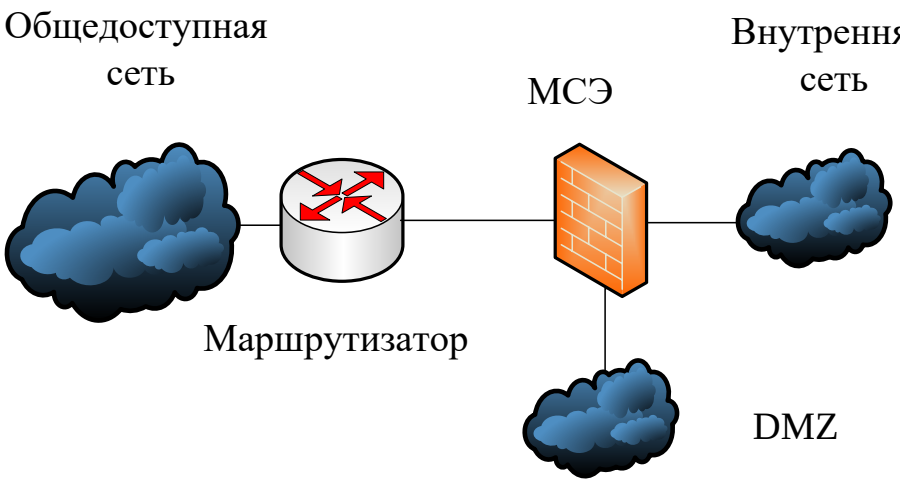


Рисунок 4.4 – Подключение МСЭ с интерфейсом для DMZ

4.2 Межсетевое экранирование с пакетной фильтрацией

Межсетевые экраны с фильтрацией пакетов представляют собой маршрутизаторы (например, Cisco) или работающие на сервере программы, сконфигурированные таким образом, чтобы фильтровать входящие и исходящие пакеты. Поэтому такие экраны называют иногда пакетными фильтрами. Фильтрация осуществляется путем анализа IP-адреса источника и получателя, а также портов входящих TCP- и UDP-сегментов и сравнением их с сконфигурированной таблицей правил. Эти межсетевые экраны просты в использовании, дешевы, оказывают минимальное влияние на производительность вычислительной системы. Основным недостатком является их уязвимость при подмене адресов IP (спуфинг). Во всей линейке оборудования Cisco Systems пакетная фильтрация реализована с помощью так называемых списков контроля доступа (Access Control List).

Списки доступа ACL могут быть созданы для всех сетевых протоколов, функционирующих на маршрутизаторе, например IP или IPX, и устанавливаются на интерфейсах маршрутизаторов. Запрет или разрешение

сетевого трафика через интерфейс маршрутизатора реализуется на основании анализа совпадения определенных условий. Для этого списки доступа представляются в виде последовательных записей, в которых используют адреса и протоколы. Сетевые фильтры (списки доступа) создаются для входящих или исходящих пакетов на основании анализируемых параметров (адреса источника, адреса назначения, протокола и номера порта верхнего уровня), указанных в списке доступа ACL (рисунок 4.5).

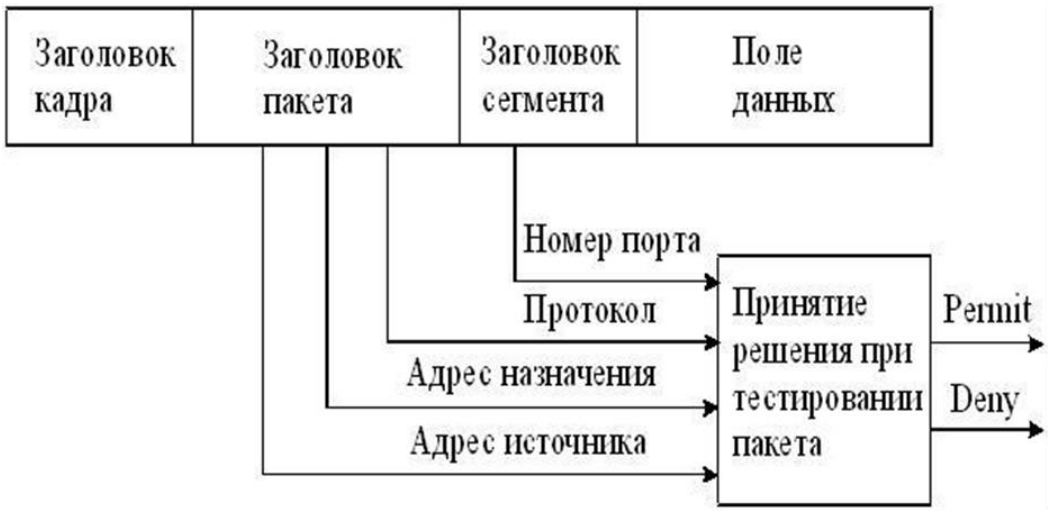


Рисунок 4.5 – Анализ заголовков пакета

Как видно из рисунка 4.5, на основе проведенного анализа служебной информации, устройство, реализующее межсетевое экранирование, принимает решение о дальнейшей передаче (permit) или о фильтрации (deny).

Списки доступа могут быть определены для каждого установленного на интерфейсе протокола и для каждого направления сетевого трафика (исходящего и входящего). Поэтому для входящего и исходящего трафиков через интерфейс создаются отдельные списки.

Если списки доступа не формируются на маршрутизаторе, то все проходящие через маршрутизатор пакеты будут иметь доступ к сети.

Список доступа ACL составляется из утверждений (условий), которые определяют, следует ли пакеты принимать или отклонять во входных и выходных интерфейсах маршрутизатора. Программное обеспечение Cisco IOS проверяет пакет последовательно по каждому условию. Если условие,

разрешающее продвижение пакета, расположено наверху списка, никакие условия, добавленные ниже его, не будут запрещать продвижение пакета. Если в списке доступа необходимы дополнительные условия, то список целиком должен быть удален и создан новый с новыми условиями.

Функционирование маршрутизатора по проверке соответствия принятого пакета требованиям списка доступа производится следующим образом. Когда кадр поступает на интерфейс, маршрутизатор проверяет IP-адрес. Если адрес назначения соответствует адресу интерфейса, то маршрутизатор извлекает (декапсулирует) из кадра пакет и проверяет его на соответствие условиям списка ACL входного интерфейса. При отсутствии запрета или отсутствии списка доступа пакет инкапсулируется в новый кадр второго уровня и отправляется интерфейсу следующего устройства.

Проверка условий (утверждений) списка доступа производится последовательно. Если текущее утверждение верно, пакет обрабатывается в соответствии с командами **permit** или **deny** списка доступа, остальная часть условий ACL не проверяется. Если все утверждения ACL неверны, то неявно заданная по умолчанию команда **deny any** (запретить все остальное) в конце списка не позволит передавать дальше по сети несоответствующие пакеты.

Существуют разные типы списков доступа: стандартные (standard ACLs), расширенные (extended ACLs) и именованные (named ACLs). Когда список доступа конфигурируется на маршрутизаторе, каждый список должен иметь уникальный идентификационный номер или уникальное имя. Номер идентифицирует тип созданного списка доступа и должен находиться в пределах определенного диапазона, заданного для этого типа списка (таблица 4.1).

Таблица 4.1 – Диапазоны идентификационных номеров ACL

Диапазон номеров	Название списка доступа
1-99	IP standard access-list
100-199	IP extended access-list
1300-1999	IP standard access-list (extended range)

2000-2699	IP extended access-list (extended range)
600-699	Appletalk access-list
800-899	IPX standard access-list
900-999	IPX extended access-list

В стандартном списке доступа для принятия решения в IP-пакете анализируется только адрес источника сообщения, чтобы фильтровать сеть (IPX-стандарт может фильтровать как адрес источника, так и назначения).

Расширенные списки доступа (Extended Access Lists) проверяют как IP-адрес источника, так и IP-адрес назначения, поле протокола в заголовке пакета сетевого уровня и номер порта в заголовке транспортного уровня.

Таким образом, для каждого протокола, для каждого направления трафика и для каждого интерфейса может быть создан свой список доступа. Исходящие фильтры не затрагивают трафик, который идет из местного маршрутизатора.

Из рекомендаций по установке списков доступа можно отметить следующее. Стандартные списки доступа рекомендуется устанавливать по возможности ближе к адресату назначения, а расширенные – ближе к источнику. Поэтому стандартные списки доступа должны блокировать устройство назначения и располагаться поближе к защищаемой сети, а расширенные списки доступа должны быть установлены близко к источнику сообщений.

Список доступа производит фильтрацию пакетов по порядку, поэтому в строках списков следует задавать условия фильтрации, начиная от специфических условий и заканчивая общими. Условия списка доступа обрабатываются последовательно от вершины списка к основанию, пока не будет найдено соответствующее условие. Если никакое условие не найдено, тогда пакет отклоняется и уничтожается, поскольку неявное условие **deny any** (запретить все остальное) присутствует неявно в конце любого списка доступа. Не удовлетворяющий списку доступа пакет протокола IP будет

отклонен и уничтожен, при этом отправителю будет послано сообщение ICMP. Новые записи (линии) всегда добавляются в конце списка доступа.

Конфигурирование списков доступа производится в два этапа:

1. Создание списка доступа в режиме глобального конфигурирования.
2. Привязка списка доступа к интерфейсу в режиме детального конфигурирования интерфейса.

Формат команды создания стандартного списка доступа следующий:

Router(config)#access-list {№} {permit / deny} {адрес источника}.

Списки доступа могут фильтровать как трафик, входящий в маршрутизатор (in), так и трафик, исходящий из маршрутизатора (out). Направление трафика указывается при привязке списка доступа к интерфейсу. Формат команды привязки списка к интерфейсу следующий:

Router(config-if)#{протокол} access-group {номер} {in или out}.

После привязки списка доступа его содержимое не может быть изменено. Не удовлетворяющий администратора список доступа должен быть удален командой **no access-list** и затем создан заново.

Расширенный список доступа создается командой:

Router(config)#access-list {№} {permit / deny} {трансп.протокол} {адр.ист} {адр.пол.} eq {№ порта или название прикладного протокола}

Правила назначения в списке доступа номера порта (или, что тоже самое, прикладного протокола) представлены в таблице 4.2.

Таблица 4.2 – Правила назначения прикладных протоколов

Обозначение	Действие
lt n	Все номера портов, меньшие n.
gt n	Все номера портов, большие n.
eq n	Порт n
neq n	Все порты, за исключением n.
range n m	Все порты от n до m включительно.

Распространенные прикладные протоколы и соответствующие им стандартные номера портов приведены в таблице 4.3.

Таблица 4.3 – Номера портов некоторых прикладных протоколов

Номер порта	Транспортный протокол	Прикладной протокол	Ключевое слово в команде access-list
20	TCP	FTP	data ftp_data
21	TCP	Управление сервером FTP	ftp
22	TCP	SSH	
23	TCP	Telnet	telnet
25	TCP	SMTP	Smtp
53	UDP, TCP	DNS	Domain
67, 68	UDP	DHCP	nameserver
69	UDP	TFTP	Tftp
80, 8080	TCP	HTTP (WWW)	www
110	TCP	POP3	pop3
161	UDP	SNMP	Snmp

Рассмотрим пример создания стандартного списка доступа для сети, схема которой показана на рисунке 4.6. На рисунке укажем узлы, находящиеся в подсетях 192.168.0.0/24 и 192.168.1.0/24.

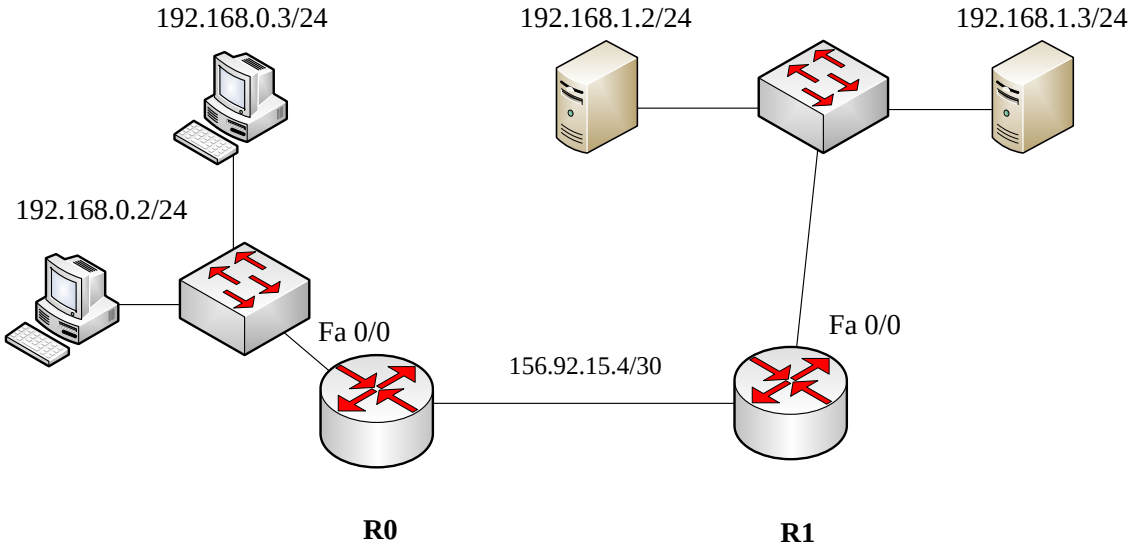


Рисунок 4.6 – Схема сети

Предположим, что к серверу, находящемуся в подсети 192.168.1.0/24 по адресу 192.168.1.2/24, доступ из подсети 192.168.0.0/24 разрешен только компьютеру 192.168.0.2/24. Это правило можно сконфигурировать с

использованием стандартного списка доступа на интерфейсе Fa 0/0 маршрутизатора R1.

Для этого в режиме глобального конфигурирования на маршрутизаторе R1 необходимо выполнить следующие команды:

Router1(config)#access-list 10 permit 192.168.0.2

Router1(config)#interface fa 0/0

Router1(config-if)#ip access-group 10 out

Первая команда создает на маршрутизаторе список доступа с номером 10, который разрешает (permit) передачу пакетов с адресом источника 192.168.0.2.

Вторая команда является командой перехода к конфигурированию интерфейса Fa 0/0.

Третья команда привязывает список доступа с номером 10 к интерфейсу Fa 0/0 и указывает на направление передачи – исходящее (out).

Созданный таким образом список доступа будет состоять из двух строк. Первая строка в явной форме разрешает передавать на интерфейс маршрутизатора Fa 0/0 пакеты с адресом источника 192.168.0.2. Вторая строка в неявном виде запрещает (deny any) передавать на этот интерфейс все остальные пакеты.

Проанализируем действия маршрутизатора R1 при поступлении на его внешний интерфейс пакета после создания списка доступа.

Если пакет поступил из подсети 156.92.15.4 и предназначен серверу 192.168.1.2, маршрутизатор, определив по таблице маршрутизации выходной интерфейс, передает этот пакет в буфер интерфейса Fa 0/0.

Затем анализируется список, начиная с первой строки. Если источник имеет адрес 192.168.0.2 (совпадение в первой строке списка произошло), пакет инкапсулируется в кадр Ethernet и передается серверу. Если источник имеет любой другой адрес (совпадения в первой строке списка не произошло), происходит обращение ко второй неявной строке списка (deny any), и пакет отбрасывается.

В случае, если необходимо обеспечить доступ к серверу и второго компьютера подсети 192.168.0.0/24 с адресом 192.168.0.3, команды конфигурирования будут выглядеть следующим образом:

Router1(config)#access-list 10 permit 192.168.0.2

Router1(config)#access-list 10 permit 192.168.0.3

Router1(config)#interface fa 0/0

Router1(config-if)#ip access-group 10 out

Очевидно, что список доступа теперь содержит три строки – две явные и одну неявную.

Очевидно, что рассмотренный способ конфигурирования списков доступа удобен в том случае, если доступ к какому-либо ресурсу (серверу) необходимо обеспечить небольшому количеству источников (компьютеров). Если же, например, в подсети 192.168.0.0/24 значительное количество компьютеров, такое конфигурирование становится неудобным и подверженным ошибкам, так как для каждого из них необходимо отдельно создавать строку списка.

Поэтому при создании списков доступа можно использовать wildcard маски. В этом случае в строке списка может содержаться указание на передачу или фильтрацию пакетов не с адресами конечных узлов, а с адресами сетей (подсетей), в которые они входят.

Правило использования масок в этом случае можно сформулировать следующим образом – нулевые значения разрядов маски означают требование обработки соответствующих разрядов адреса, а единичные значения разрядов маски означают игнорирование соответствующих разрядов адреса. Например, если wildcard маска имеет вид 0.0.0.0, то проверять условие необходимо для всех разрядов адреса источника прибывшего пакета. Если же маска имеет вид 0.0.0.255, то проверять условие необходимо только для первых трех байтов адреса источника.

Предположим, что доступ к тому же серверу (адрес 192.168.1.2) должны получить все компьютеры подсети 192.168.0.0/24. В этом случае на маршрутизаторе R1 необходимо выполнить команды:

Router1(config)#access-list 10 permit 192.168.0.0 0.0.0.255

Router1(config)#interface fa 0/0

Router1(config-if)#ip access-group 10 out

Необходимо отметить, что, если нужно разрешить какому-либо одному узлу из другой подсети (например, 192.168.2.2/24) доступ к этому же серверу, создаваемый список необходимо дополнить командой

Router1(config)#access-list 10 permit 192.168.2.2 0.0.0.0

или, что то же самое, командой

Router1(config)#access-list 10 permit host 192.168.2.2

Создание списков доступа очень похоже на создание «белых» и «черных» списков на телефоне. При создании «белого» списка принимать разрешено только вызовы от источников, номера которых внесены в «белый» список, остальные вызовы отбрасываются. При использовании «черного» списка отбрасываются только вызовы от источников, внесенных в список, остальные вызовы принимаются. Основное отличие от телефонных вызовов состоит в том, что в списки **permit** и **deny** вносятся не телефонные номера, а значения заголовков различных уровней.

Используя эту аналогию с «белыми» и «черными» списками телефона, можно отметить, что рассмотренные способы аналогичны созданию в телефоне «белых» списков – указанные в списке доступа адреса являются разрешенными, остальные – запрещенными.

В ряде случаев более удобным является использование аналогии «черного» списка – разрешено передавать данные от всех, кроме тех, кто указан в черном списке.

Предположим, что к тому же серверу необходимо обеспечить доступ всем компьютерам, кроме одного, имеющего адрес 192.168.0.15. Конфигурирование такого списка будет иметь вид:

Router1(config)#access-list 11 deny host 192.168.0.15

Router1(config)#access-list 11 permit any

Router1(config)#interface fa 0/0

Router1(config-if)#ip access-group 11 out

Напомним, что по умолчанию у создаваемых списков доступа неявно присутствует заключительная строка **deny any** – запретить все. В данном случае мы заменили эту строку на **permit any** – разрешить все. Соответственно, доступ к серверу будет разрешен всем, кроме компьютера с адресом 192.168.0.15.

Рассмотрим теперь применение расширенного списка для конфигурирования маршрутизатора R1 (рисунок 4.6), при этом должны быть выполнены следующие условия:

- компьютеру 192.168.0.2/24 необходимо предоставить доступ к web-серверу с адресом 192.168.1.2 по протоколу WWW;

- всем компьютерам подсети 192.168.0.0/24 необходимо предоставить доступ к FTP-серверу с адресом 192.168.1.3 по протоколу FTP.

Команды конфигурирования в этом случае будут выглядеть следующим образом:

Router1(config)#access-list 110 permit tcp host 192.168.0.2 host 192.168.1.2 eq www

Router1(config)#access-list 110 permit tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp

Router1(config)#interface fa 0/0

Router1(config-if)#ip access-group 110 out

Очевидно, что указанный способ аналогичен созданию «белого» списка в телефоне, так как третье неявное условие, находящееся в конце списка, блокирует все, что не разрешено.

Рассмотрим пример, когда удобнее использовать аналогию «черного» списка в телефоне.

На маршрутизаторе R1 должны быть выполнены следующие условия:

- компьютеру 192.168.0.2/24 необходимо запретить доступ к серверу с адресом 192.168.1.2 по протоколу WWW, но разрешить доступы к другим сервисам;

- всем компьютерам подсети 192.168.0.0/24 необходимо запретить доступ к серверу с адресом 192.168.1.3 по протоколу FTP, но разрешить доступ к другим сервисам.

Команды конфигурирования в этом случае будут иметь вид:

```
Router1(config)#access-list 110 deny tcp host 192.168.0.2 host 192.168.1.2 eq www
Router1(config)#access-list 110 deny tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp
Router1(config)#access-list 110 permit ip any any
Router1(config)#interface fa 0/0
Router1(config-if)#ip access-group 110 out
```

Запись **permit ip any any** означает, что весь остальной трафик от любого источника к любому получателю должен передаваться.

Просмотреть созданные на маршрутизаторе списки доступа можно по команде **show access-list**, а списки, настроенные на конкретных интерфейсах, командами **show ip interfaces** или **show running-config**. На рисунке 4.7 показаны настроенные списки доступа для рассмотренного здесь примера.

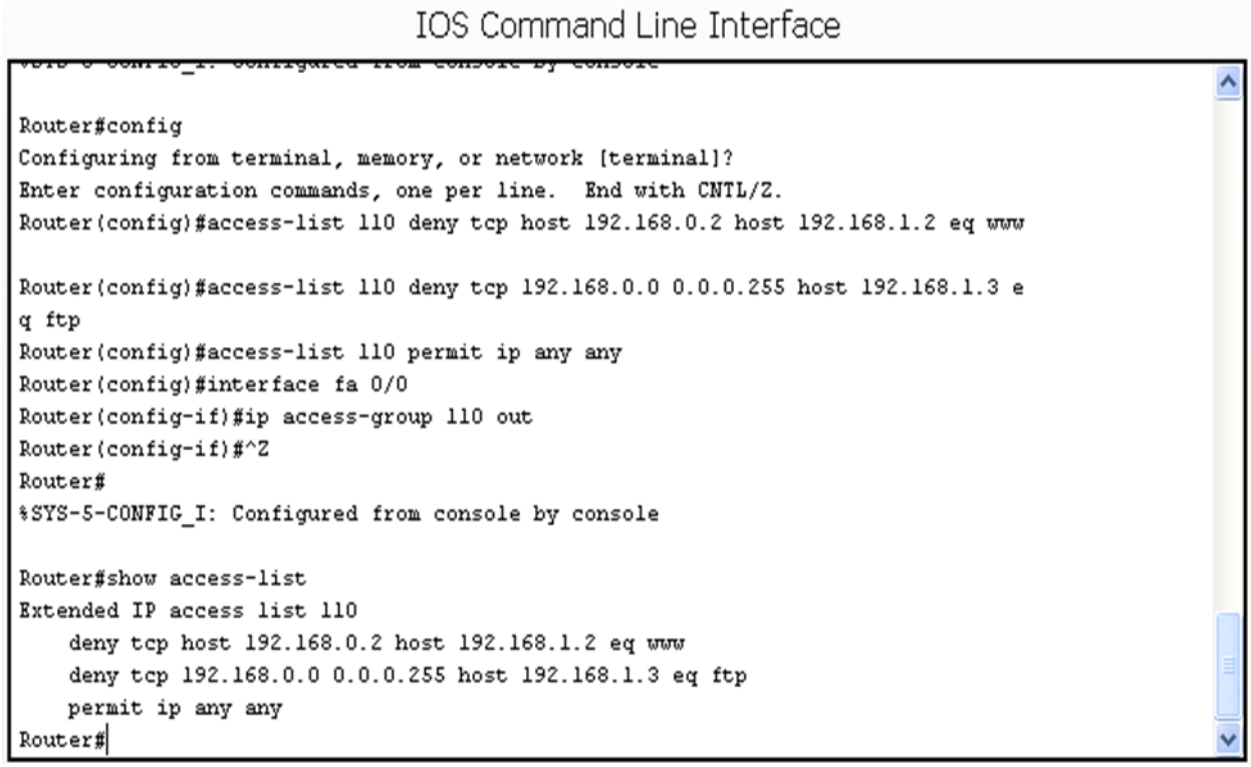


Рисунок 4.7 – Конфигурирование списка доступа и его просмотр

Списки доступа также желательно использовать и для конфигурирования удаленного доступа к устройствам.

В настоящее время чаще используются не нумерованные, а именованные списки доступа. Удобство использования именованных списков доступа заключается прежде всего в том, что названию списка можно придать определенный смысл (INTERNET, ADMIN, FTP, и т.д.). Так как именованный список не имеет номера, который однозначно определяет его вид (таблица 3.1), при создании такого списка необходимо явно указать, какой именно список создается – стандартный или расширенный. Команда создания именованного списка доступа имеет вид:

ip access-list <standard/extended> <имя>

<правило 1>

<правило 2>

<правило n>

Параметр **standard/extended** указывает на вид создаваемого списка, а правила прописываются аналогично нумерованным спискам.

4.3 Межсетевое экранирование с сохранением состояний

Такие межсетевые экраны еще называют экранами с сохранением сессий (statefull firewall), а саму технологию межсетевого экранирования – SPI (Stateful Packet Inspection). Они относятся к шлюзам сеансового уровня (параграф 4.1) Суть заключается в том, что при запросе на установление соединения (например, TCP-сессии) маршрутизатор запоминает эту сессию и при поступлении извне пакета сверяет его со всеми текущими сессиями. Если принятый извне пакет относится к какой-либо текущей сессии, он продвигается во внутреннюю сеть, в противном случае – отбрасывается.

Для конфигурирования межсетевого экранирования на устройствах Cisco необходимо в явном виде указать, трафик каких протоколов должен отслеживаться (инспектироваться). Для этого используется команда

ip inspect name <имя правила> <название протокола>

Данная команда выполняется в режиме глобального конфигурирования.

Аналогично спискам доступа, созданное правило необходимо привязать к интерфейсу с указанием направления передачи:

R1(config)#int fa0/0 – переход в режим конфигурирования интерфейса fa 0/0;

R1(config-if)#ip inspect <имя правила> <in/out> - привязка правила к интерфейсу с указанием направления передачи.

Необходимо отметить, что правило можно привязывать как к внутреннему, так и ко внешнему интерфейсу маршрутизатора, однако направление передачи должно соответствовать направлению запросов из внутренней сети ко внешней. Соответственно, если правило привязывается к внутреннему интерфейсу, направление передачи – входящее (in), если к внешнему – исходящее (out).

Приведем пример межсетевого экранирования для сети, показанной на рисунке 4.8. Для удобства разместим маршрутизатор R1 во внешней сети, в которой также располагаются два сервера – web-сервер и ftp-сервер.

Произведем настройку всего оборудования таким образом, чтобы из внутренней сети были доступны оба сервера.

Создадим правило для инспектирования запросов к web-серверу (протокол HTTP) с именем HTTP и привяжем его к внутреннему интерфейсу fa0/0 с входящим направлением передачи (рисунок 4.9).

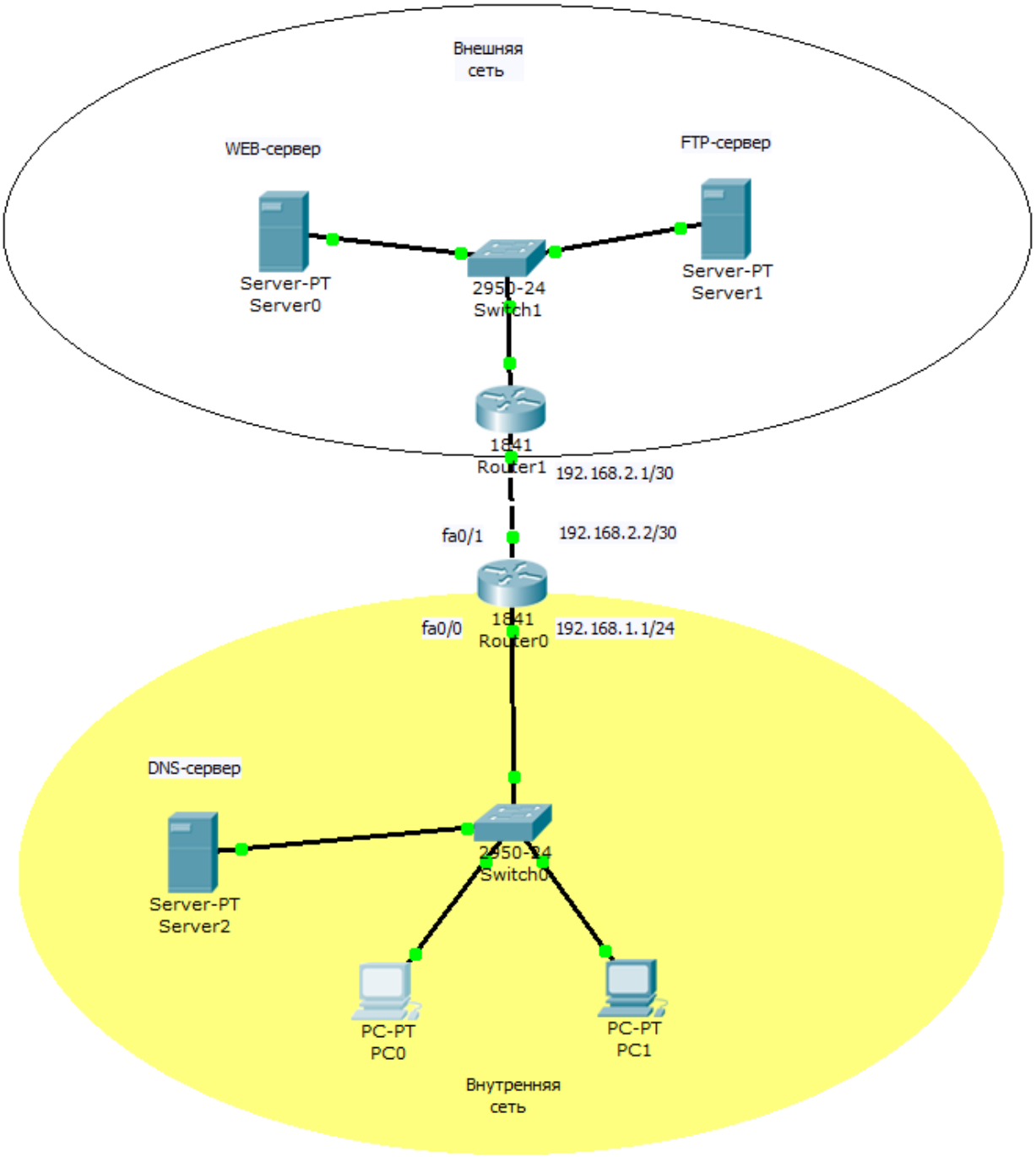


Рисунок 4.8 – Пример сети

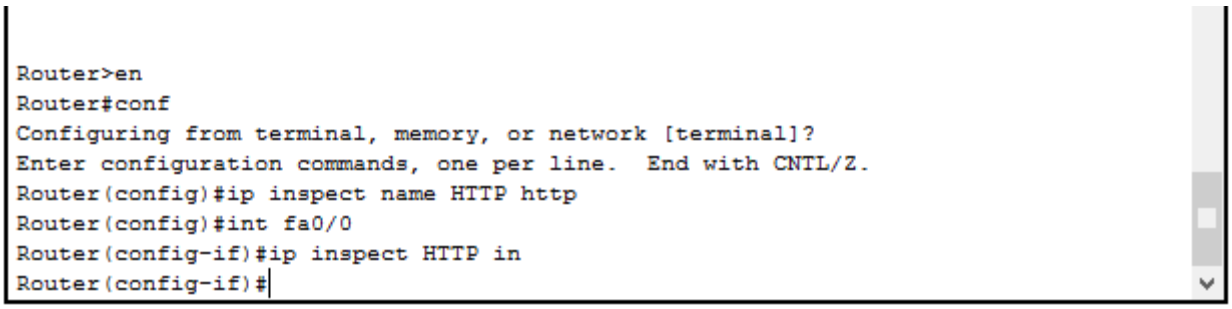


Рисунок 4.9 – Конфигурирование инспектирования протокола HTTP

Следует отметить (и это очень важно!), что инспектирование трафика необходимо применять совместно со списками доступа. В нашем примере, когда списки доступа не были созданы, http-запросы, поступающие из внутренней сети, будут инспектироваться. Однако если из внешней сети также поступит http-запрос, он пройдет во внутреннюю сеть, так как не существует списка доступа, обеспечивающего фильтрацию этого запроса.

Для проверки этого разместим во внешней сети ПК с адресом 213.80.65.4 и попробуем соединиться с сервером внутренней сети по протоколу HTTP (рисунок 4.10).

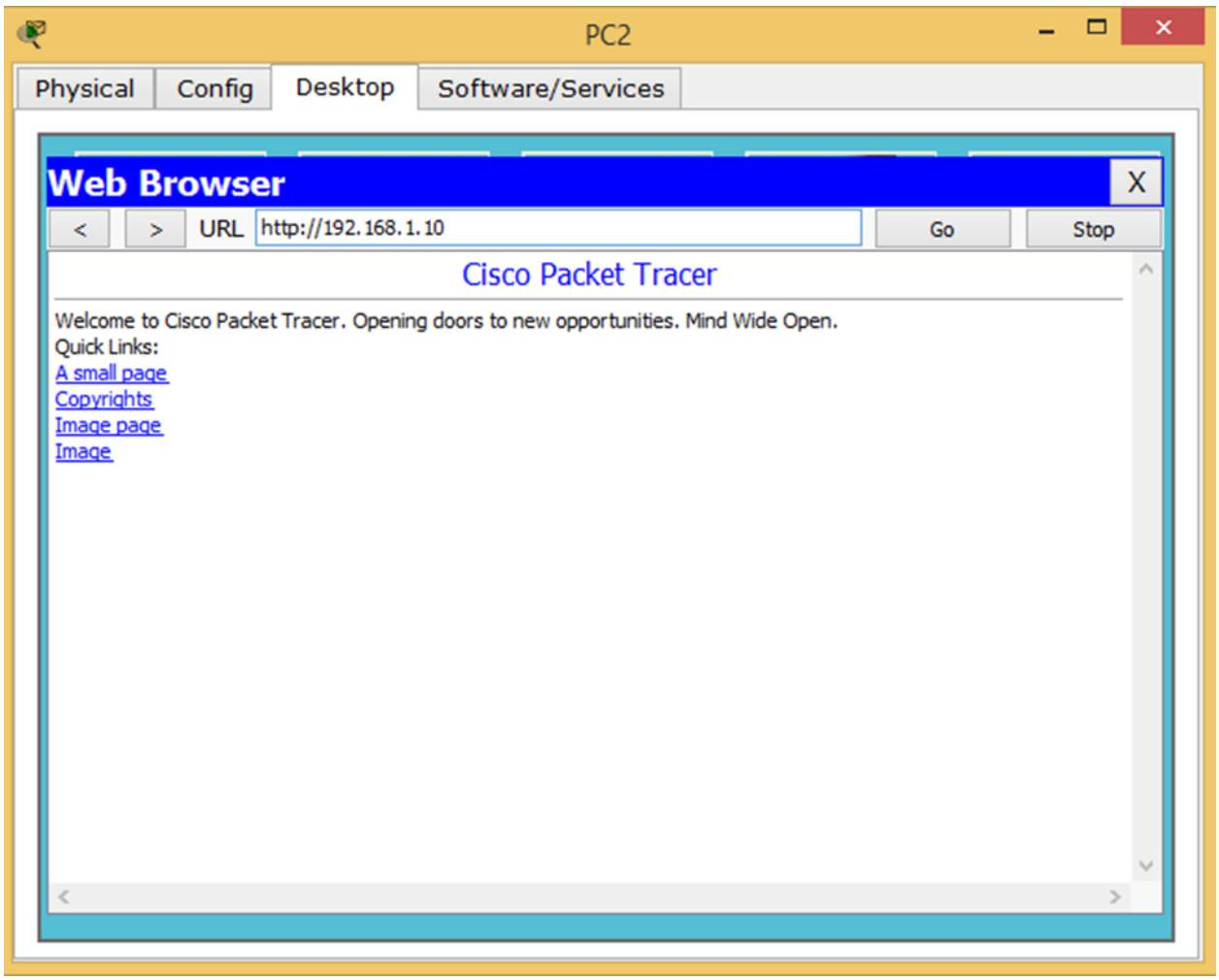


Рисунок 4.10 – Соединение с сервером внутренней сети по протоколу HTTP

В качестве внутреннего сервера мы использовали DNS-сервер с адресом 192.168.1.10/24. Как видно из рисунка 4.10, соединение прошло успешно.

Для защиты внутренней сети создадим на маршрутизаторе R0 список доступа, запрещающий передачу всех IP-пакетов, и привяжем его к внешнему интерфейсу fa0/1 с указанием входящего направления (рисунок 4.11).

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip access-list ex FRW
Router(config-ext-nacl)#deny ip any any
Router(config-ext-nacl)#
```

Рисунок 4.11 – Создание списка доступа

Теперь снова попытаемся послать HTTP-запрос из внешней сети (рисунок 4.12).

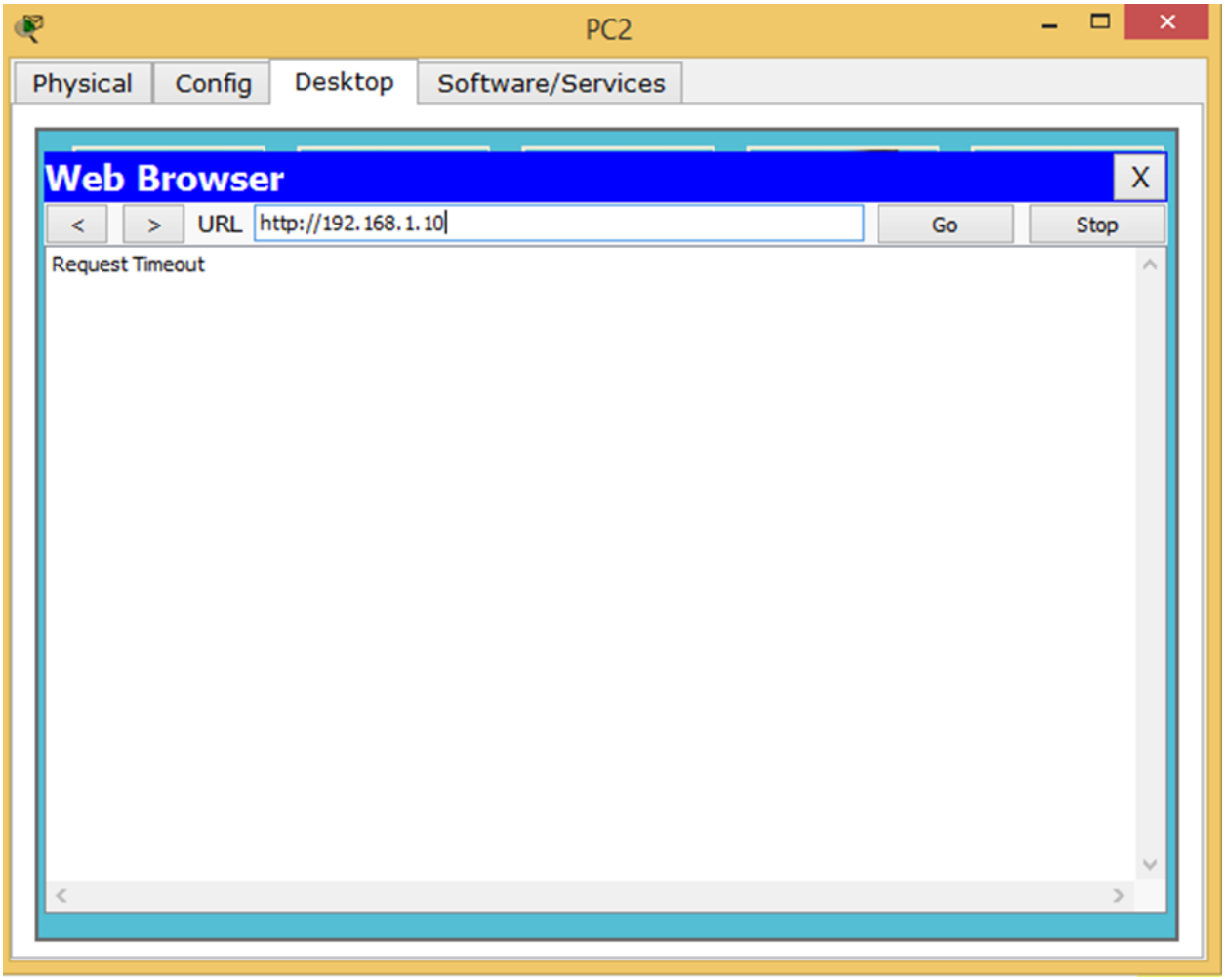


Рисунок 4.12 – Отсутствие соединения с сервером внутренней сети по протоколу HTTP

Как видно из рисунка 4.12, из внешней сети сервер не доступен. При этом из внутренней сети web-сервер остается доступным.

С учетом созданного списка доступа FRW остальные протоколы (кроме HTTP) не инспектируются. Следовательно, если послать из внутренней сети запрос по какому-либо другому протоколу, ответ получен не будет, так как его заблокирует список доступа на внешнем интерфейсе маршрутизатора. Попробуем получить из внутренней сети доступ к ftp-серверу (рисунок 4.13).

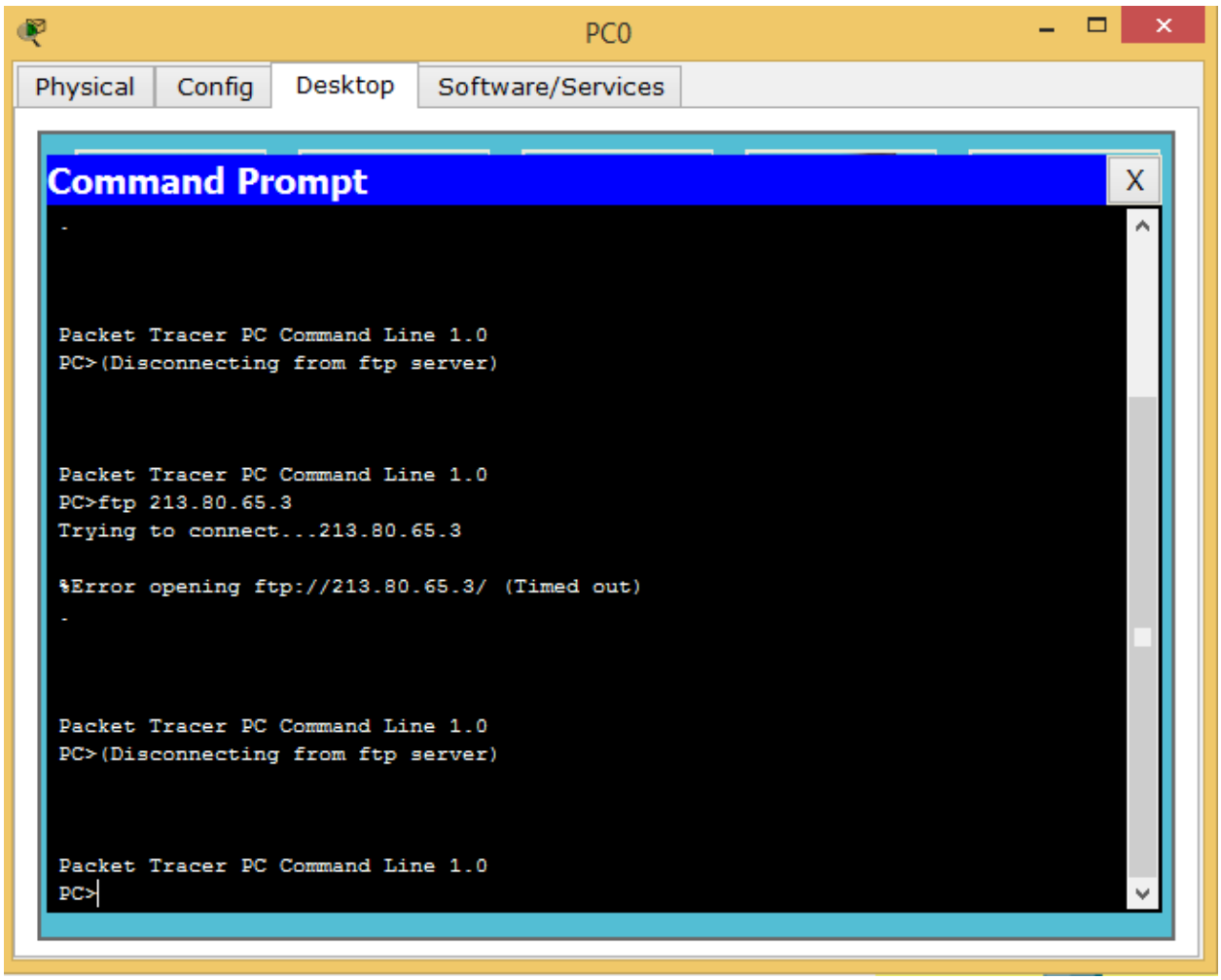


Рисунок 4.13 – Попытка соединения с ftp-сервером

Как видим, попытка не увенчалась успехом. Если же настроить инспектирование FTP-трафика, доступ к ftp-серверу будет возможен. Иллюстрировать здесь это не будем, так как Cisco Packet Tracer в силу своей ограниченной функциональности это не поддерживает. Поэтому инспектирование разных видов трафика необходимо производить либо на реальном оборудовании, либо с использованием симулятора GNS3.

Однако Cisco Packet Tracer поддерживает инспектирование TCP-трафика. Так как и протокол HTTP, и протокол FTP использует для передачи TCP-сегменты, после настройки TCP-инспектирования доступны окажутся и http и ftp серверы (рисунки 4.14, 4.15).

```
Router(config)#ip inspect name HTTP tcp
Router(config)#
Router(config)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Рисунок 4.14 – Настройка инспектирования TCP-трафика

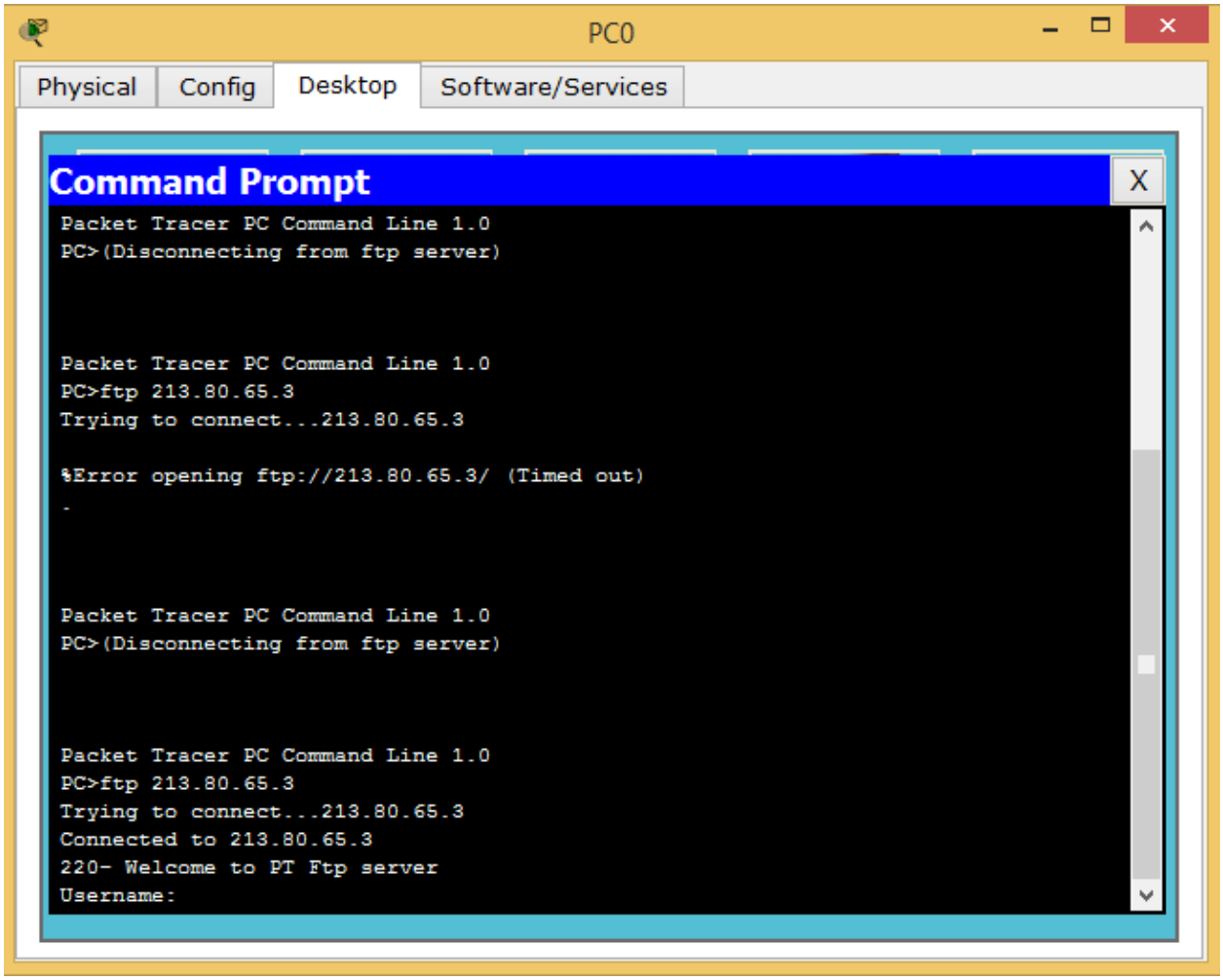


Рисунок 4.15 – Доступ к ftp-серверу после настройки инспектирования TCP-трафика

Рассмотренные настройка межсетевого экрана являются базовыми, а более тонкие настройки применяются в случае, если производится разделение сети с различными зонами безопасности, которые рассмотрим в следующем параграфе.

4.4 Zone-Based Policy Firewall (ZBFW)

Zone-Based Policy Firewall (ZBFW) – относительно новое направление на маршрутизаторах под управлением операционной системы Cisco IOS для конфигурирования правил доступа между сетями. До появления этой технологии трафик фильтровался с помощью списков доступа ACL и динамической инспекции трафика Context-Based Access Control (CBAC) (в настоящем пособии не рассматривается). И ACL и правила CBAC применяются непосредственно на физические интерфейсы, что во многих случаях не способствует масштабируемости и гибкости сетевых решений. Такая модель ограничивает степень детализации политик межсетевого экрана и вызывает путаницу правильного применения политики межсетевого экранирования, особенно в случаях, когда политика межсетевого экрана должна применяться между несколькими интерфейсами. Zone-Based Policy Firewall меняет конфигурацию межсетевого экрана от старой интерфейсной модели на более гибкую модель, основанную на зонах безопасности.

Зона безопасности состоит из набора различных интерфейсов, которые должны иметь одинаковую политику сетевой безопасности или, иначе говоря, одинаковый уровень доверия. В каждую из зон может входить один или несколько интерфейсов. После создания зон настраиваются правила для взаимодействий между зонами. Такой подход облегчает настройки правил межсетевого экрана, так как правила определяются не для отдельных интерфейсов, а для множества интерфейсов, входящих в одну зону. Кроме того в Zone-Based Policy Firewall используется язык Cisco Policy Language (CPL), который позволяет более гибко, чем в предыдущих версиях межсетевого экрана, настраивать правила фильтрации трафика.

В большинстве случаев сеть делится, как минимум, на три зоны:

- внутренняя зона, где расположены пользователи (inside);
- внешняя зона (Интернет – outside);
- демилитаризированная зона, где расположены серверы, к которым должен быть обеспечен доступ извне (dmz).

Важно, что по умолчанию весь трафик между различными зонами будет запрещен, весь трафик внутри зоны – разрешен.

На первом этапе настройки межсетевого экрана необходимо создать зоны. Зоны создаются командой, выполняемой в режиме глобального конфигурирования:

zone security <имя зоны>

После этого необходимо создать пары зон, между которыми будет передаваться трафик:

zone-pair security <имя пары> source <имя зоны> destination <имя зоны>

Необходимо отметить, что пары зон являются однонаправленными. То есть, если в нашей сети предполагается двунаправленная передача данных между внутренней и внешней зонами, необходимо создать две пары зон. Например (считаем, что внутренней зоне было присвоено имя IN, а внешней – OUT, имя пар IN_OUT и OUT_IN):

zone-pair security IN_OUT source IN destination OUT

zone-pair security OUT_IN source OUT destination IN

Как указывалось выше, по умолчанию передача трафика между созданными парами зон запрещена. Для формирования разрешенного для передачи трафика необходимо определить критерии, по которым отсортировывается нужный трафик. Для этого используется так называемый class-map (дословно – классовая карта). Class-map определяет, какой именно трафик будет инспектироваться (проходить между зонами, также проходить будут ответы на этот трафик). Фильтроваться трафик может по критериям с 3-го по 7-й уровней модели OSI (т.е. начиная от IP-адреса и заканчивая трафиком определенного приложения или сервиса прикладного уровня). Определяться трафик может списками доступа, значением CoS, типом протокола и еще рядом других параметров. Критериев может присутствовать одновременно несколько. При этом можно указать, должен ли трафик попадать под все эти критерии (match-all) или под любой из них (match-any).

Таким образом, основная задача class-map – отфильтровать необходимый тип трафика.

Для создания class-map используется команда:

class-map type inspect match-all/match-any <имя class-map>

После этого мы попадаем в конфигурирование созданного class-map, и далее необходимо использовать команду **match** с указанием критериев, по которым отсортировывается трафик:

- **access-group** - стандартный, расширенный или именованный список доступа, который может фильтровать трафик на основании IP адреса и порта источника и приемника. Это единственный способ выделить трафик от конкретного источника к конкретному получателю;

- **protocol** - это протоколы уровня 4 (TCP, UDP, ICMP), а также прикладные сервисы, такие как HTTP, SMTP, DNS, и т.д. Может быть указан любой известный или определяемый пользователем сервис;

- **class-map** - подчиненный класс, который предоставляет дополнительные критерии соответствия;

- **not** - определяет, что любой трафик, который не соответствует указанному сервису или протоколу, или листу доступа, будет выбран в данном class-map.

Важно, что критерии вводятся списком, и порядок обработки списка последовательный, как и у списков доступа. Например, если при конфигурировании class-map match-any мы использовали команды

match protocol http

match protocol tcp

то при обработке пакета сначала будет проверено его соответствие протоколу HTTP. Если найдено соответствие, то далее будет инспектироваться этот трафик, и следующее условие не будет проверяться. Если же команды поменять местами, то пакет сначала попадет под инспектирование трафика TCP.

Политики межсетевого экранирования определяются командой **policy-map**. Команда **policy-map** определяет действие, которое будет произведено с отфильтрованным с помощью команды **class-map** трафиком. Существует три основных действия, которые применимы к классифицированному трафику:

Drop – Трафик, обрабатываемый этим действием, отбрасывается и никакого уведомления на удаленный хост не высылается (в противоположность классическим листам доступа (ACL), когда высылается ICMP-сообщение Host Unreachable). Каждая карта политик имеет скрытый класс class-default, для которого сконфигурировано действие **Drop** (аналогично строке deny any any в любом списке доступа).

Pass – Пропускает трафик, не включая инспекцию протокола. Это действие позволяет маршрутизатору пересылать трафик из одной зоны в другую, при этом он не отслеживает состояние соединений или сессий. Это действие разрешает прохождение трафика только в одном направлении. Чтобы обратный трафик был передан, должна быть соответствующая политика и для него. Это действие полезно для таких протоколов, как IPSec ESP, IPSec AH, ISAKMP и других по своей сути безопасных протоколов с предсказуемым поведением.

Inspect - Включает динамическую инспекцию для трафика, который проходит от зоны источника к зоне приемника, и автоматически разрешает обратный трафик даже для сложных протоколов, таких как H.323. Например, если трафик передается из зоны IN в зону OUT, маршрутизатор поддерживает информацию о соединениях или сеансах для TCP и UDP трафика. Поэтому маршрутизатор разрешает обратный трафик из зоны OUT в зону IN в качестве ответов на запросы соединений из IN в OUT.

Формат команды:

policy-map type inspect <имя policy-map >

После этого мы попадаем в режим конфигурирования созданного policy-map, в котором указываем, какой именно class-map должен обрабатываться, и затем указываем необходимое действие:

class type inspect <имя class-map>

inspect/pass/drop

Теперь созданные политики необходимо применить к парам зон, которые уже были созданы ранее (пары зон можно создать и на этом шаге):

zone-pair security <имя пары> source <имя зоны> destination <имя зоны>

service-policy type inspect <имя policy-map >

Осталось в явном виде указать маршрутизатору, какие его интерфейсы относятся к какой зоне:

interface <имя интерфейса>

zone-member security <имя зоны>

Приведем пример конфигурирования межсетевого экранирования на примере сети, показанной на рисунке 4.16.

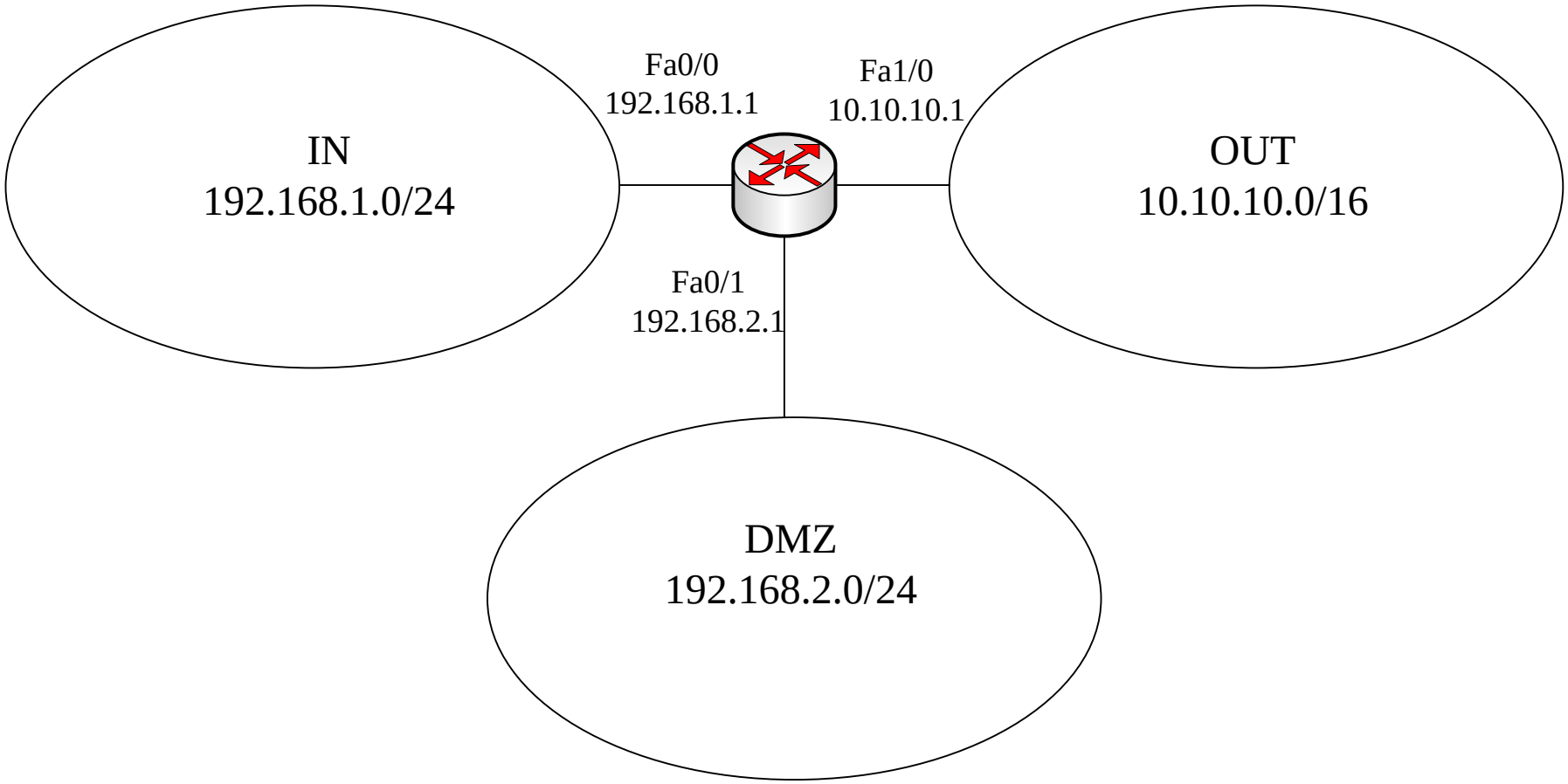


Рисунок 4.16 – Пример сети

Для простоты будем считать, что интерфейсам маршрутизатора присвоены первые адреса из адресного пространства подсетей (192.168.1.1/24 в подсети IN, 10.10.10.1/16 в подсети OUT, и т.д.)

Сначала необходимо определить зоны:

R0(config)#zone security IN

R0(config)#zone security OUT

R0(config)#zone security DMZ

Так как созданные зоны пока не привязаны ни к каким интерфейсам, никакого ограничения в передаче трафика после создания зон не произойдет.

Сначала сконфигурируем межсетевой экран для информационного обмена между зонами IN и OUT. Считаем, что из внутренней сети разрешены любые запросы к серверу, находящемуся во внешней сети. Таким образом, при создании class-map в этом направлении удобно использовать стандартный список доступа, разрешающий передавать данные от всех рабочих станций подсети 192.168.1.0/24:

R0(config)#access-list permit 10 192.168.1.0 0.0.0.255

Создаем class-map для трафика, удовлетворяющего условию списка доступа 10, присвоив самому class-map имя FROM-IN:

R0(config)#class-map type inspect match-any FROM-IN

и указываем, какой трафик входит в созданный class-map:

R0(config-cmap)#match access-group 10

Создаем policy-map с именем FROM-INSIDE (в принципе, имена class-map и policy-map могут совпадать, здесь специально выбраны разные имена):

R0(config)#policy-map type inspect FROM-INSIDE

указываем, какой class-map должна обрабатывать политика:

R0(config-pmap)#class type inspect FROM-IN

и указываем нужное действие – инспектировать:

R0(config-pmap-c)#inspect

Теперь обращаемся к интерфейсам для указания, к каким зонам они относятся. Одновременно можно назначить интерфейсам IP-адреса и включить их, если этого не было сделано раньше:

R0(config)#int fa0/0

R0(config-if)#ip address 192.168.1.1 255.255.255.0

R0(config-if)#no shutdown

R0(config-if)#zone-member security IN

R0(config)#int fa1/0

R0(config-if)#ip address 10.10.10.1 255.255.0.0

R0(config-if)#no shutdown

R0(config-if)#zone-member security OUT

Так как интерфейсы маршрутизатора теперь принадлежат к разным зонам, передача трафика между ними запрещена. Для разрешения передачи между ними трафика в соответствии с созданной политикой создадим зонную пару с именем IN-TO-OUT:

R0(config)#zone-pair security IN-TO-OUT source IN destination OUT

и применим к ней созданную политику:

R0(config-sec-zone-pair)#service-policy type inspect FROM-INSIDE

Заметим, что пару для обратного направления OUT-IN мы не создавали. Это связано с тем, что по условиям задачи любой трафик извне запрещен, за исключением ответов на запросы, которые поступили из внутренней сети (они инспектируются). Проверить работоспособность сконфигурированного межсетевого экрана достаточно просто – если из внутренней сети послать любой запрос (например, ping или http-запрос), то внутренний компьютер должен получить ответ (рисунок 4.17). Если же послать запрос с внешнего сервера, ответа не будет – запрос будет отброшен маршрутизатором (рисунок 4.18).

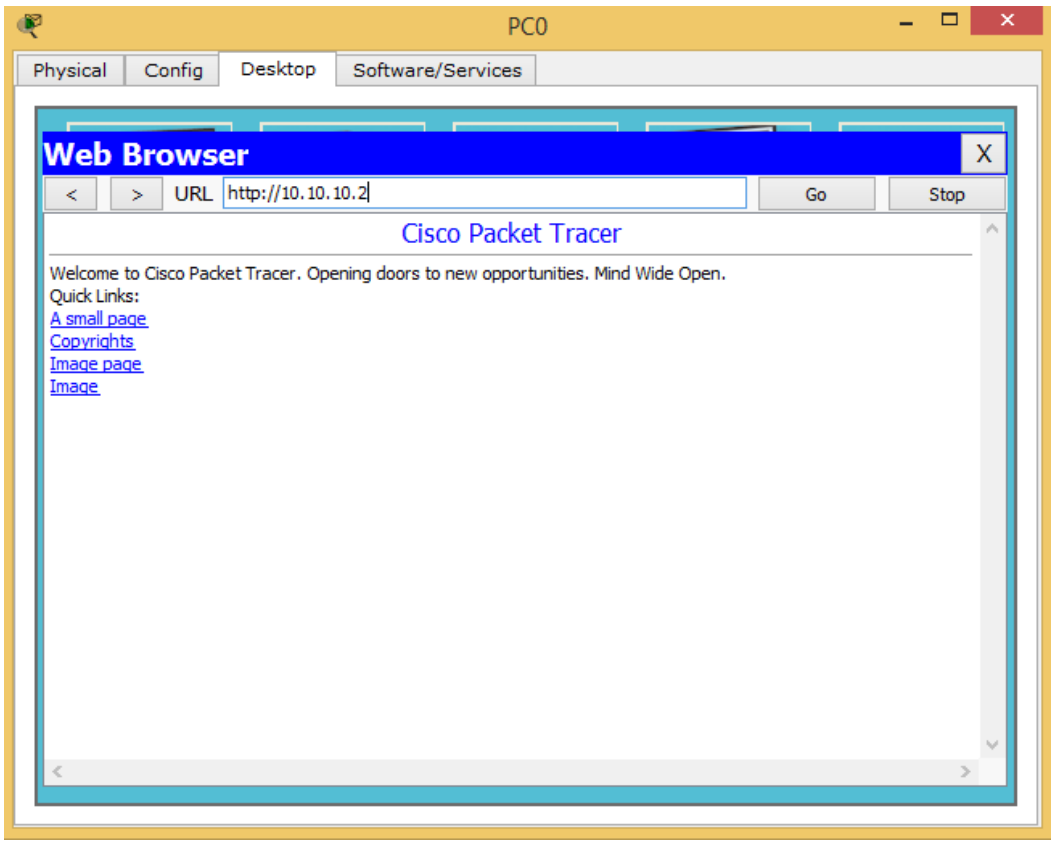


Рисунок 4.17 – Получение ответа при запросе из внутренней сети

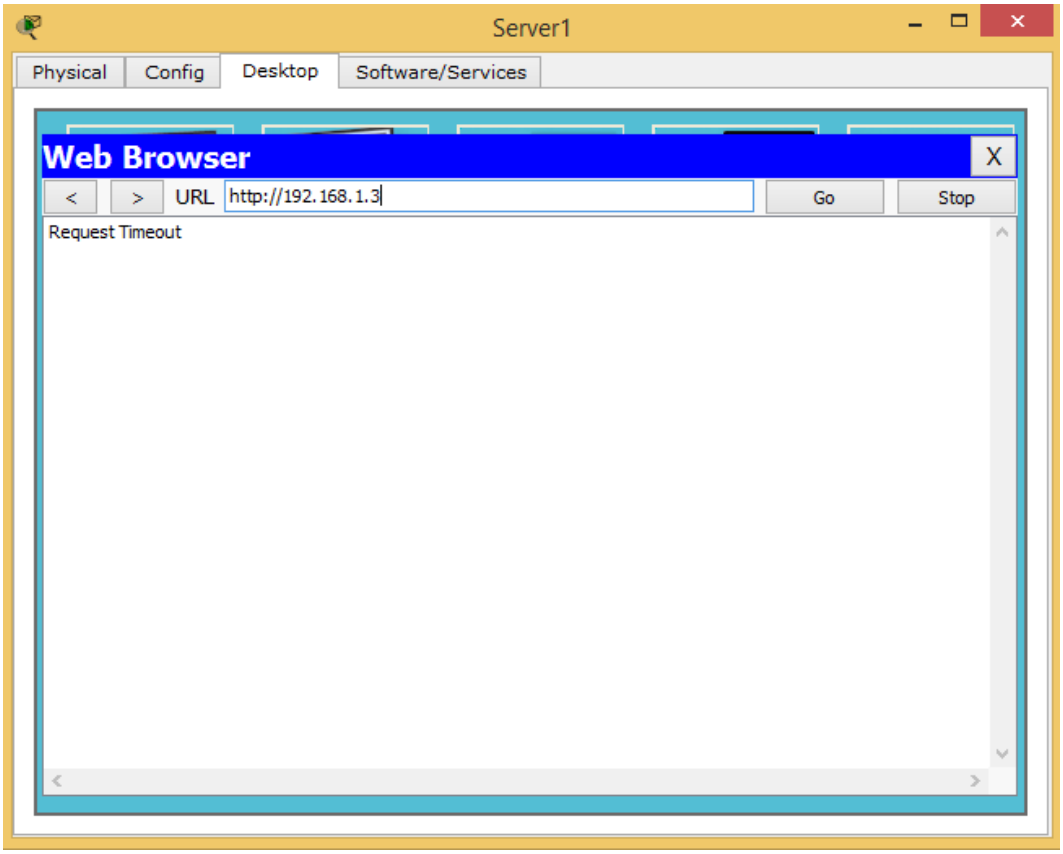


Рисунок 4.18 – Отсутствие ответа при запросе из внешней сети

Перейдем теперь к настройке демилитаризованной зоны. Здесь ситуация несколько иная – к серверам демилитаризированной должен быть обеспечен доступ как извне (например, по протоколу HTTP, если это web-сервер), так и изнутри (например, по протоколу SSH или Telnet для конфигурирования сервера). Соответственно, в этом случае необходимо создавать как минимум две пары зон – OUT-DMZ, IN-DMZ – с различными политиками. Если же предполагается наличие доступа из демилитаризованной зоны, необходимо создавать пары DMZ-OUT или DMZ-IN.

Предположим, что для нашей сети необходимо обеспечить следующие условия:

1. Доступ из внешней сети к серверу, находящемуся в DMZ, возможен только по протоколу HTTP;

2. Доступ из внутренней сети к серверу, находящемуся в DMZ, возможен по протоколам SSH, Telnet, HTTP, и только с тех IP-адресов, которые относятся к адресному пространству внутренней сети 192.168.1.0/24.

Привяжем интерфейс маршрутизатора fa0/1 к созданной ранее зоне DMZ, одновременно назначим ему IP-адрес и включим его:

R0(config)#int fa0/1

R0(config-if)#ip address 192.168.2.1 255.255.255.0

R0(config-if)#no shutdown

R0(config-if)#zone-member security DMZ

Очевидно, что для передачи трафика из внутренней сети к серверу DMZ необходимо создать несколько class-map, так как в каждом из них одновременно должно выполняться как минимум два условия:

- источником трафика является внутренняя сеть и используется протокол SSH;

- источником трафика является внутренняя сеть и используется протокол Telnet;

- источником трафика является внутренняя сеть и используется протокол HTTP.

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу SSH с именем IN-DMZ-SSH:

R0(config)#class-map type inspect match-all IN-DMZ-SSH

R0(config-cmap)#match access-group 10

R0(config-cmap)#match protocol ssh

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу Telnet с именем IN-DMZ-TLN:

R0(config)#class-map type inspect match-all IN-DMZ-TLN

R0(config-cmap)#match access-group 10

R0(config-cmap)#match protocol telnet

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу HTTP с именем IN-DMZ-HTTP:

R0(config)#class-map type inspect match-all IN-DMZ-HTTP

R0(config-cmap)#match access-group 10

R0(config-cmap)#match protocol http

Создадим policy-map с именем IN-DMZ, в которой укажем на необходимость инспектирования трафика, удовлетворяющего созданным class-map:

R0(config)#policy-map type inspect IN-DMZ

R0(config-pmap)#class type inspect IN-DMZ-SSH

R0(config-pmap-c)#inspect

R0(config-pmap-c)#exit

R0(config-pmap)#class type inspect IN-DMZ-TLN

R0(config-pmap-c)#inspect

R0(config-pmap-c)#exit

R0(config-pmap)#class type inspect IN-DMZ-HTTP

R0(config-pmap-c)#inspect

R0(config-pmap-c)#exit

R0(config-pmap)#

Для доступа к серверу DMZ из внешней сети должен использоваться только протокол HTTP, поэтому создадим один class-map с именем OUT-DMZ:

R0(config)#class-map type inspect match-any OUT-DMZ

R0(config-cmap)#match protocol http

Создадим policy-map с таким же названием и с указанием инспектировать трафик:

R0(config)#policy-map type inspect OUT-DMZ

R0(config-pmap)#class type inspect OUT-DMZ

R0(config-pmap-c)#inspect

Осталось создать пары зон и применить к ним созданные политики. Пара IN-TO-DMZ:

R0(config)#zone-pair security IN-TO-DMZ source IN destination DMZ

R0(config-sec-zone-pair)#service-policy type inspect IN-DMZ

Пара зон OUT-TO-DMZ:

R0(config)#zone-pair security OUT-TO-DMZ source OUT destination DMZ

R0(config-sec-zone-pair)#service-policy type inspect OUT-DMZ

4.5 Практическое задание

4.5.1 Межсетевое экранирование с пакетной фильтрацией

4.5.1.1 В Cisco Packet Tracer создать сеть, состоящую из пяти подсетей. В одной из подсетей установить два сервера, один из которых должен быть сконфигурирован как FTP, а другой – как WEB-сервер.

4.5.1.2 Всем компьютерам подсети 192.169.X.0 (где X – номер студента в списке группы) предоставить полный доступ ко всем серверам.

4.5.1.3 Всем компьютерам подсети 192.169.X+1.0 предоставить доступ только к FTP-серверу по протоколу FTP.

4.5.1.4 Всем компьютерам подсети 192.169.X+2.0 предоставить доступ только к WEB-серверу.

4.5.1.5 Компьютерам оставшейся подсети запретить доступ к внешним ресурсам.

4.5.2 Межсетевое экранирование с сохранением состояний

4.5.2.1 Используя Cisco Packet Tracer, построить сеть, показанную на рисунке 4.8. Произвести конфигурирование сетевых устройств для обеспечения доступа всех серверов из внутренней сети.

4.5.2.2 Убедиться в возможности доступа во внутреннюю сеть извне с использованием произвольного протокола.

4.5.2.3 Настроить инспектирование TCP-трафика и сконфигурировать список доступа. Убедиться в доступности серверов из внутренней сети и недоступности ресурсов внутренней сети извне.

4.5.3 Zone-Based Policy Firewall (ZBFW)

4.5.3.1 Собрать сеть, показанную на рисунке 4.16, настроить адресацию по следующим исходным данным:

- внутренняя сеть – 192.168.X.0;

- демилитаризованная зона – 192.168.X+10.0;

- внешняя сеть – 213.80.X.0.

4.5.3.2 Проверить связность сети.

4.5.3.3 Сконфигурировать на маршрутизаторе межсетевой экран ZBFW, обеспечивающий выполнение следующих правил:

- из внутренней сети разрешены все запросы к внешней сети;

- к демилитаризованной зоне разрешены запросы из внутренней сети по протоколам Telnet и SSH и только с адресов, принадлежащих внутренней сети;

- из внешней сети запросы во внутреннюю сеть запрещены;

- из внешней сети запросы в демилитаризованную зону разрешены только по протоколу HTTP.

4.5.3.4 Проверить работоспособность межсетевого экрана.

5 Использование дополнительных средств защиты локальных сетей

5.1 Технология NAT

Выше уже упоминалась технология трансляции сетевых адресов – Network Address Translation (NAT). NAT широко используется в современных сетях по следующим причинам. Во-первых, уже сейчас наблюдается дефицит IP-адресов четвертой версии. Кардинальным решением здесь может служить переход к шестой версии IP-протокола, но пока повсеместно используется IPv4. При использовании NAT в пределах внутренней сети могут использоваться частные адреса, о которых уже шла речь в третьей главе настоящего пособия. Преобразование частных адресов в общедоступные и обратно осуществляется с использованием протокола NAT. Одни и те же частные адреса могут использоваться в различных корпоративных сетях, что и приводит к экономии адресного пространства.

Во-вторых, NAT существенно повышает безопасность корпоративной сети, так как в этом случае извне сеть представляется единственным или несколькими общедоступными адресами. Поэтому определить структуру корпоративной сети, проанализировать данные, циркулирующие в ней, становится проблематично.

Основная идея технологии NAT состоит в следующем [1]. Внутренняя корпоративная сеть использует адресное пространство частных адресов. В маршрутизаторе или другом устройстве, связывающем внутреннюю сеть с внешней IP-сетью, настраивается протокол NAT, осуществляющий при передаче во внешнюю сеть преобразование частного адреса в общедоступный и обратное преобразование при приеме. Так как внутренняя сеть также может содержать маршрутизаторы для разделения ее на подсети, они должны получать объявления о маршрутной информации от маршрутизаторов внешней сети. В свою очередь, внешние маршрутизаторы не должны ничего знать о маршрутизаторах внутренней сети. Поэтому NAT-устройство должно пропускать из внешней сети во

внутреннюю сообщения протоколов маршрутизации (RIP, OSPF и т.д.), но не пропускать эти сообщения в обратном направлении. Число общедоступных адресов чаще всего меньше числа частных адресов, за счет чего и достигается экономия адресного пространства. В частном, но далеко не самом редком случае, может использоваться всего один общедоступный адрес, настраиваемый на внешнем порту NAT-маршрутизатора.

Адресное пространство частных адресов представлено в таблице 5.1.

Таблица 5.1 – Адресное пространство частных адресов

Адрес сети	Маска подсети	Диапазон адресов
10.0.0.0	255.0.0.0	10.0.0.0 – 10.255.255.255
172.16.0.0 – 172.31.0.0	255.240.0.0	172.16.0.0 – 172.31.255.255
192.168.0.0	255.255.0.0	192.168.0.0 – 192.168.255.255

Рассмотрим сначала наиболее простой случай, когда количество конечных узлов внутренней сети равно количеству общедоступных адресов, полученных данной сетью от провайдера сетевых услуг (рисунок 5.1).

На рисунке представлены две внутренние сети, обозначенные A и B, связанные между собой через общедоступную сеть. Выход из внутренней сети в общедоступную осуществляется с использованием NAT-устройства, в качестве которого может использоваться маршрутизатор или межсетевой экран с установленным программным обеспечением NAT. В данном примере полагаем, что внутренняя адресация каждой из сетей одинакова, то есть и в сети A, и в сети B могут быть узлы с одинаковыми частными IP-адресами (192.168.1.1 в данном примере).

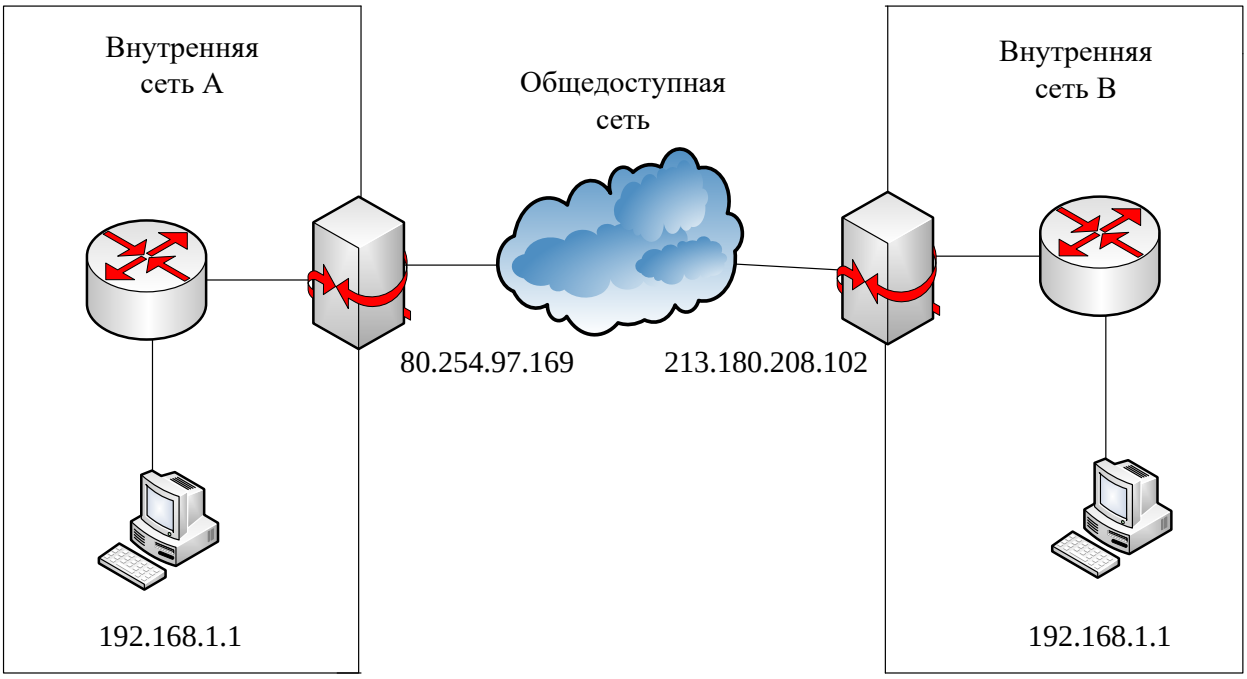


Рисунок 5.1 – Простейший случай использования NAT

Внешние адреса NAT-устройств являются общедоступными и, соответственно, уникальными.

Предположим, что конечный узел внутренней сети А собирается послать пакет данных конечному узлу внутренней сети В. В качестве IP-адреса получателя в пакете указывается адрес 213.180.208.102, и пакет передается на маршрутизатор внутренней сети А. Как указывалось выше, внутренние маршрутизаторы получают уведомления о маршрутной информации из внешней сети, поэтому внутренний маршрутизатор сети А «знает» о маршруте к адресу 213.180.208.102, в нашем примере этот маршрут пролегает через NAT-устройство. Соответственно, пакет попадает на NAT-устройство, соединяющее внутреннюю сеть А с общедоступной сетью.

Однако в пакет должен быть помещен также и IP-адрес отправителя. Конечный узел сети А помещает в пакет свой адрес – 192.168.1.1, и этот пакет без каких-либо изменений достигает NAT-устройства сети А. В свою очередь, NAT-устройство должно подменить адрес источника 192.168.1.1 на свой общедоступный адрес 80.254.97.169 (точнее, на адрес своего

внешнего интерфейса). Эта подмена осуществляется с использованием таблицы, хранящейся в памяти NAT-устройства, упрощенный вид которой представлен в таблице 5.1.

Таблица 5.1 – Соответствие частных и общедоступных адресов

Частный адрес	Общедоступный адрес
192.168.1.1	80.254.97.169

Очевидно, что количество общедоступных адресов у NAT-устройства должно соответствовать количеству узлов внутренней сети, имеющих права доступа во внешнюю сеть.

Пакет с измененным адресом источника достигает NAT-устройства сети B, которое хранит в своей памяти аналогичную таблицу 5.2.

Таблица 5.2 – Соответствие частных и общедоступных адресов

Частный адрес	Общедоступный адрес
192.168.1.1	213.180.208.102

Приняв данный пакет, NAT-устройство сети B изменяет адрес получателя в пакете в соответствии с таблицей 5.2, то есть адрес 213.180.208.102 изменяется на адрес 192.168.1.1. Видоизмененный таким образом пакет передается на внутренний маршрутизатор сети B и в конечном итоге достигает нужного узла.

В частном случае, когда сеть B не использует технологию NAT, пакет передается в узел сети B без изменений.

Рассмотренный пример использования NAT имеет ряд существенных недостатков.

Во-первых, экономии адресов в данном случае не происходит – внутренние адреса жестко закреплены за общедоступными адресами в таблицах NAT-устройств. Поэтому в этом виде NAT может использоваться только для повышения безопасности сети.

Во-вторых, записи в таблицу в данном случае являются статическими, то есть их необходимо вносить вручную, что при значительном количестве внутренних узлов является трудоемкой процедурой, подверженной ошибкам. Однако следует заметить, что иногда статические записи в таблице NAT необходимы, например, если во внутренней сети имеется сервер, к которому нужно обеспечить доступ из внешней сети.

Соответственно, рассмотренный выше NAT получил название статического NAT.

Для преодоления указанных недостатков был разработан динамический NAT, суть которого рассмотрим с использованием рисунка 5.2.

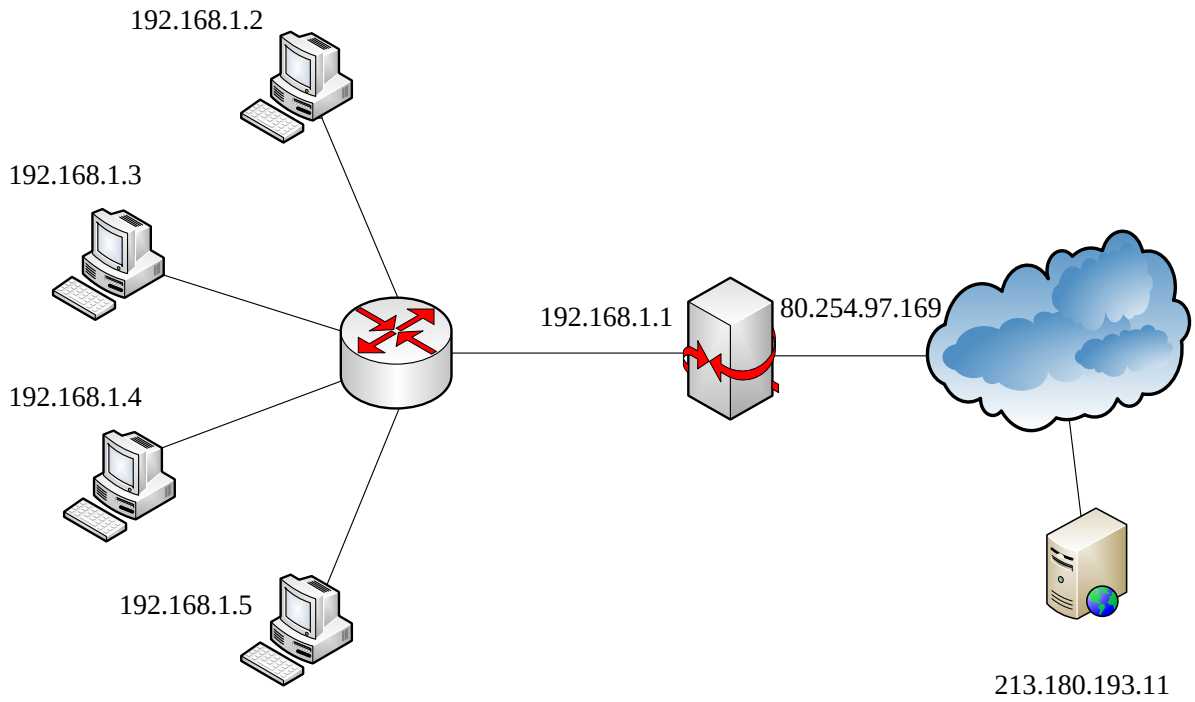


Рисунок 5.2 – Иллюстрация работы динамического NAT

На рисунке представлена внутренняя сеть, использующая частный адрес 192.168.1.0/24. Выход во внешнюю сеть организуется с использованием NAT-устройства, внешнему интерфейсу которого присвоен общедоступный адрес 80.254.97.169. Необходимо обеспечить всем четырем

конечным узлам внутренней сети доступ к внешней сети, в частности, к web-серверу с адресом 213.180.193.11.

Очевидно, что статический NAT для решения такой задачи непригоден, так как доступ узлов к внешней сети осуществляется с использованием единственного внешнего адреса (на практике внешних адресов также может быть несколько, но в любом случае количество внутренних узлов превышает количество внешних адресов).

При передаче пакета во внешнюю сеть NAT-устройство может подменить частный адрес отправителя на свой общедоступный адрес, как и в статическом NAT. Однако при приеме пакета-ответа из внешней сети необходимо определить, какому из внутренних конечных узлов этот пакет нужно передать. Или, другими словами, при приеме необходимо определить, на какой частный адрес нужно изменить общедоступный адрес назначения, содержащийся в ответном IP-пакете.

Таким образом, для надежного различения принимаемых пакетов NAT-устройством необходима, помимо IP-адресов, дополнительная информация. В качестве такой информации можно использовать номера портов TCP- или UDP-сегментов, переносимых IP-пакетами. Однако в нашем примере все четыре узла могут обратиться с запросом к web-серверу с адресом 213.180.193.11, ответы которого будут иметь один и тот же номер порта 80 или 8080. Поэтому в данном случае используются так называемые назначенные номера портов. В качестве назначенных портов используются порты источника, которым в процессе передачи присваиваются значения, не стандартизированные в протоколах TCP и UDP. Назначенный порт может быть выбран произвольно, но с учетом того, что он должен быть уникален в пределах внутренней сети.

В соответствии с этим таблица NAT-устройства усложняется, в нее теперь должны входить не только IP-адреса, но и номера портов (таблица 5.3).

Поскольку описанная выше технология использует не только сетевые адреса, но и номера портов, она получила название NAPT (Network Address Port Translation) [5].

Таблица 5.3 – Соответствие адресов и номеров портов

Частный адрес	Порт	Общедоступный адрес	Назначенный порт
192.168.1.2	8080	80.254.97.169	61001
192.168.1.3	8080	80.254.97.169	61002
192.168.1.4	8080	80.254.97.169	61003
192.168.1.5	8080	80.254.97.169	61004

При передаче пакета, например, от узла 192.168.1.2 к серверу глобальной сети с адресом 213.180.193.11 в заголовок пакета в качестве адреса получателя будет указан 213.180.193.11, в качестве номера порта получателя – 8080. В качестве адреса отправителя будет указан 192.168.1.2, а в качестве номера порта отправителя – 8080. После приема этого пакета NAT-устройством будет произведена подмена адреса отправителя на 80.254.97.169, а номера порта отправителя на 61001. Эта информация динамически заносится в таблицу 5.3.

При приеме ответа от сервера глобальной сети будет выполнено обратное преобразование – адрес получателя будет заменен на 192.168.1.2. При этом в качестве номера порта получателя будет указан назначенный порт, который сервер укажет исходя из номера порта источника принятого сегмента. При этом NAT-устройство «поймет», какому из внутренних узлов передать пакет, используя номер назначенного порта.

Если NAT-устройство имеет несколько общедоступных адресов (пул адресов), то таблица 5.3 ведется динамически, то есть при передаче пакета запоминается, на какой именно адрес из пула была осуществлена подмена, и данная информация заносится таблицу. Эти действия, естественно, являются абсолютно прозрачными для конечных узлов.

Рассмотрим настройку протокола NAT для примера, представленного на рисунке 5.2, полагая, что в качестве NAT-устройства используется маршрутизатор Cisco.

Предположим, что в маршрутизаторе, используемом в качестве NAT-устройства, порт с адресом 192.168.1.1 является портом fa 0/0, а порт с адресом 80.254.97.169 – портом fa 0/1 (напомним, что в устройствах и программном обеспечении Cisco Systems fa означает Fast Ethernet). В терминологии NAT порт fa 0/0 является внутренним портом (inside), а порт fa 0/1 – внешним портом (outside).

Пакеты, прибывающие на внутренний порт и подлежащие передаче на внешний порт, подлежат трансляции в соответствии с Source NAT (SNAT), то есть подмене подлежит IP-адрес источника (Source IP). Пакеты, прибывающие на внешний порт, подлежат трансляции в соответствии с Destination NAT (DNAT), то есть подмене подлежит IP-адрес получателя (Destination IP).

Сначала необходимо создать список доступа. Для этого в режиме глобального конфигурирования необходимо выполнить следующую команду:

(config)# access-list 100 permit ip <адрес> <wildcard mask> any

Данной командой создан список доступа с номером 100, разрешающий передавать пакеты с адресом источника, указанного в команде, на любые адреса. Список нужен, как и при межсетевом экранировании, для выделения трафика.

Пул адресов создается на маршрутизаторе в режиме глобального конфигурирования командой

(config)# ip nat pool <имя> <начальный адрес> <конечный адрес> netmask <маска>

Если, как в нашем примере, используется единственный общедоступный адрес, начальный и конечный адреса в команде совпадают.

Затем назначаются внутренние и внешние интерфейсы:

(config)# interface fa 0/0;

(config-if)# ip nat inside (outside).

Включается NAT командой

ip nat inside source list 100 pool <имя>

Конфигурирование маршрутизатора Cisco с использованием указанных команд для нашего примера (рисунок 5.2) представлен на рисунке 5.3.

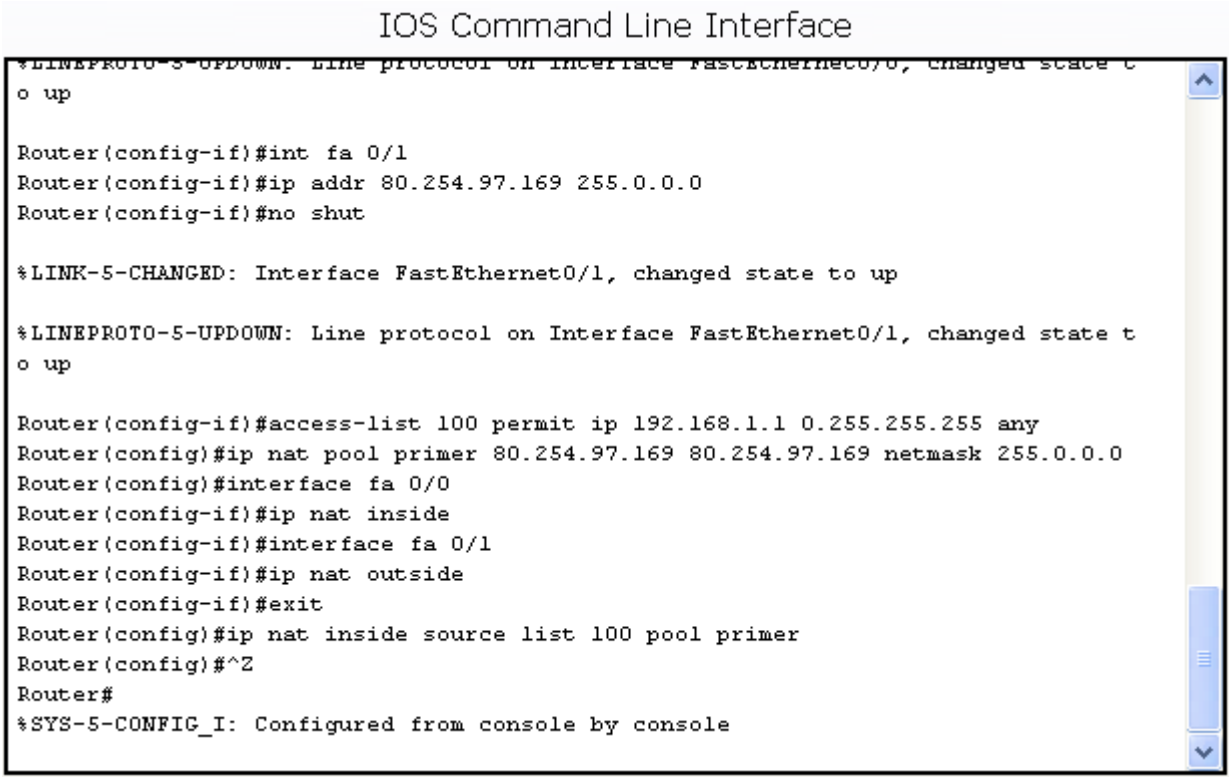


Рисунок 5.3 – Конфигурирование динамического NAT

После того, как какой-либо из внутренних узлов обменяется пакетами с внешней сетью, можно будет просмотреть трансляции адресов, произведенные NAT (рисунок 5.4).

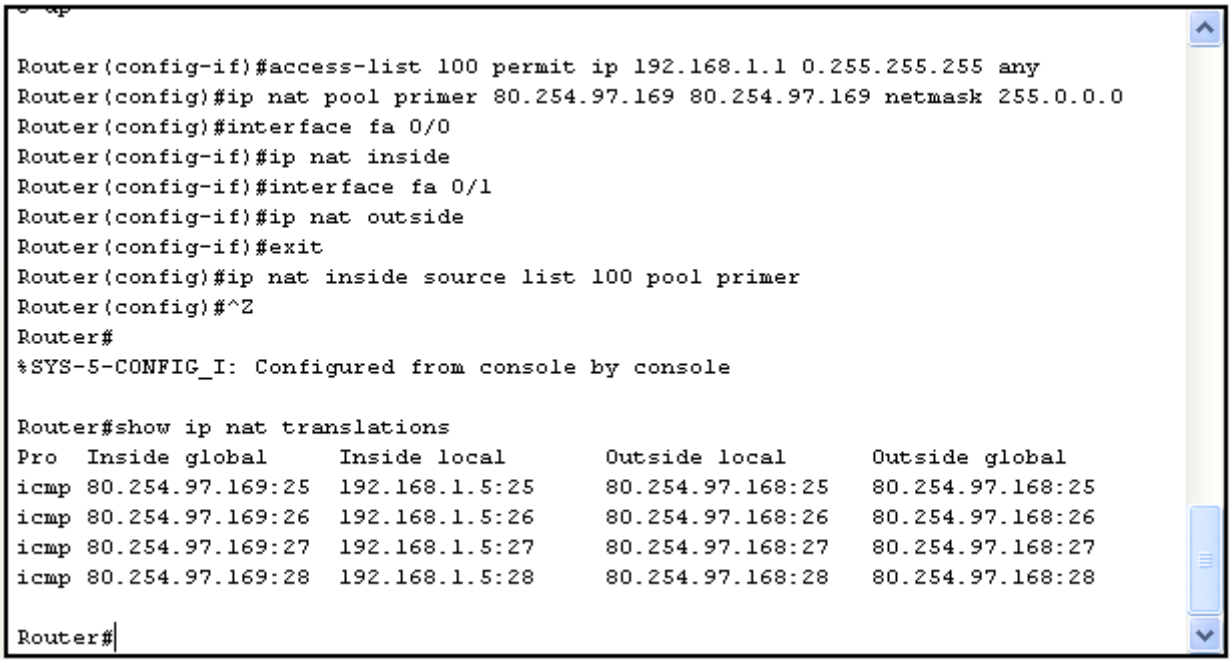


Рисунок 5.4 – Список трансляций

Для большей наглядности произведем обращение с внутреннего компьютера к web-серверу, расположенному во внешней сети по адресу 80.254.97.168, и опять выведем список трансляций (рисунок 5.5).

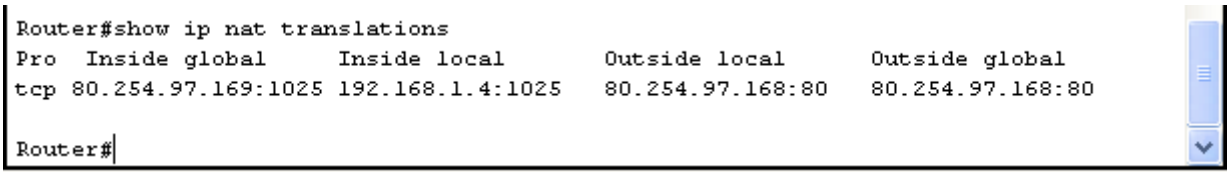


Рисунок 5.5 – Список трансляций после обращения к web-серверу

Из рисунка 5.5 следует, что была произведена одна трансляция, информация о которой представлена в четырех колонках.

Первая колонка указывает на транспортный протокол, в нашем случае это TCP.

Адреса local – это адреса, используемые при передаче по внутренней сети. Таким образом, Inside local – это сокет отправителя, Outside local – сокет получателя. Адреса global – это адреса, используемые при передаче по общедоступной сети. Outside global – это сокет получателя

(он не изменился при трансляции), Inside global – это сокет отправителя (был подвергнут трансляции).

Таким образом, из рисунка 5.5 следует, что внутренний узел с адресом 192.168.1.4 направляет пакет web-серверу с адресом 80.254.97.168. Соответственно, IP-адрес и номер порта получателя, указанные в пакете:

80.254.97.168:80 (напомним, что для протокола HTTP используются порты 80 и 8080).

IP-адрес и порт источника в этом же пакете:

192.168.1.4:1025.

При передаче пакета во внешнюю сеть маршрутизатор подменяет IP-адрес:

80.254.97.169:1025.

Порт источника остался прежним, но если бы он оказался занят, он тоже был бы подвергнут трансляции.

Соответственно, при приеме ответного пакета от сервера сокет 80.254.97.169:1025 будет изменен на 192.168.1.4:1025, и пакет получит нужный узел внутренней сети.

В заключение параграфа отметим, что здесь показаны только базовые настройки протокола NAT, на практике используется гораздо большее число настроек. Однако, получив первоначальные навыки с использованием материала, представленного в данном параграфе, можно освоить и другие возможности, предоставляемые NAT.

5.2 Применение виртуальных локальных сетей (VLAN)

Практически все современные управляемые коммутаторы поддерживают технологию виртуальных локальных сетей (VLAN). Данная технология позволяет разделить порты одного или нескольких коммутаторов по различным сетям, изолированным друг от друга.

Причин такого разделения единой сети на виртуальные сети несколько.

Во-первых, сети на коммутаторах слабо защищены от так называемого широковещательного шторма (Broadcast storm). Под широковещательным штормом понимается наводнение сети широковещательными кадрами, то есть кадрами, у которых в адресе назначения одни единицы (DA=F::F). Отказаться от широковещательных кадров нельзя, на их использовании основаны многие протоколы. Широковещательный шторм может быть спровоцирован некорректной работой сетевого оборудования, неправильной настройкой протокола покрывающего дерева STP (данный протокол будет рассмотрен ниже), а также действиями злоумышленников.

Во-вторых, при разделении сети легче локализовать возникающие проблемы, проще и быстрее можно устранить возникшие неисправности.

В-третьих, с использованием виртуальных сетей легче разграничить права доступа к ресурсам сети, хотя эту задачу решают не только VLAN, а еще ряд технологий и протоколов.

Технология VLAN описана в открытом стандарте IEEE 802.1Q [9], кроме того, существует фирменный стандарт Cisco Systems, называемый ISL.

Реализация технологии VLAN на одном коммутаторе предельно проста – при разделении портов между разными виртуальными сетями для каждой из них как бы создается своя адресная таблица (на практике адресная таблица остается единой, но в ней указывается номер VLAN). При реализации VLAN на нескольких коммутаторах используется дополнительное поле (tag), которое используется для передачи кадров между коммутаторами или маршрутизаторами.

Однако на практике необходимым бывает обеспечение данных между различными виртуальными сетями. Это обеспечивается при помощи устройств третьего уровня – маршрутизаторов и коммутаторов L3.

Рассмотрим сначала настройку маршрутизации между VLAN с использованием маршрутизатора на примере сети, показанной на рисунке 5.6.

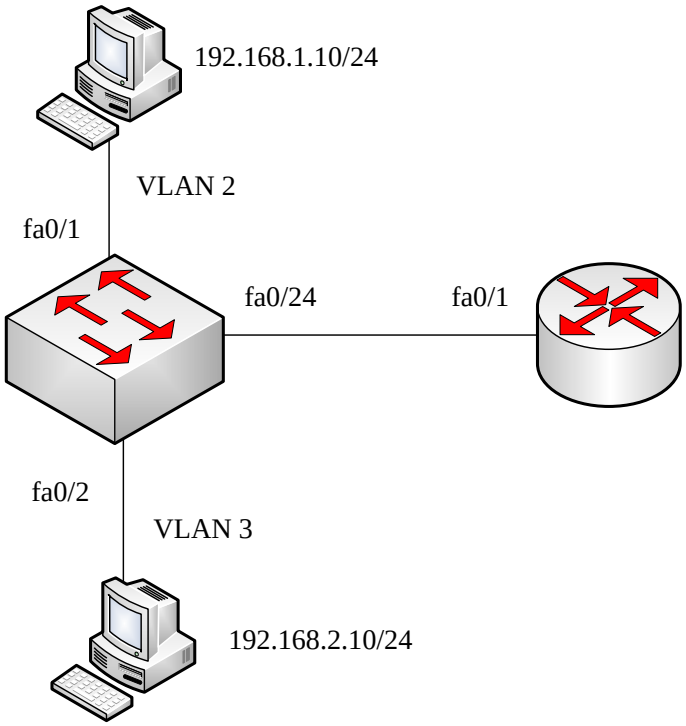


Рисунок 5.6 – Пример сети

Как видно из рисунка, на коммутаторе организованы две VLAN. Соответственно, трафик между ними запрещен. Команды конфигурирования VLAN на коммутаторе:

Switch(config)#vlan 2

Switch(config-vlan)#name vlan_number_2

Switch(config-vlan)#exit

Switch(config)#interface fastEthernet 0/1

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 2

Switch(config-if)#exit

Switch(config)#vlan 3

Switch(config-vlan)#name vlan_number_3

Switch(config-vlan)#exit

Switch(config)#interface fastEthernet 0/2

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 3

Switch(config-if)#exit

Переведем порт fa0/24 в транковый режим и разрешим передавать через него трафик VLAN 2 и 3:

Switch(config)#interface fastEthernet 0/24

Switch(config-if)#switchport mode trunk

Switch(config-if)#switchport trunk allowed vlan 2-3

Команды конфигурирования маршрутизатора:

Router(config)#interface fastEthernet 0/1

Router(config-if)#no shutdown

Router(config)#interface fastEthernet 0/1.2

Router(config-subif)#encapsulation dot1Q 2

Router(config-subif)#ip address 192.168.1.1 255.255.255.0

Router(config)#interface fastEthernet 0/1.3

Router(config-subif)#encapsulation dot1Q 3

Router(config-subif)#ip address 192.168.2.1 255.0.0.0

Router(config-subif)#exit

Команда **encapsulation dot1Q 2** означает, что кадры, исходящие из этого виртуального интерфейса, будут иметь ID второй VLAN. Остальные команды уже были рассмотрены ранее.

Такой способ маршрутизации между VLAN называется «маршрутизатор на привязи» и имеет ряд недостатков.

Во-первых, при большом числе VLAN весь трафик передается по единому физическому соединению – в нашем примере между портами fa0/24 коммутатора и fa0/1 маршрутизатора. Это может элементарно перегрузить соединение, то есть не хватит пропускной способности.

Во-вторых, маршрутизаторы работают гораздо медленнее коммутаторов. Благодаря тому, что коммутация осуществляется аппаратно, а не программно, коммутация осуществляется практически на скорости канала.

В-третьих, стоимость порта маршрутизатора всегда выше стоимости порта коммутатора, то есть решение «маршрутизатор на привязи» достаточно дорогое.

Рассмотрим теперь настройку маршрутизации между VLAN с использованием коммутатора третьего уровня на примере сети, показанной на рисунке 5.7.

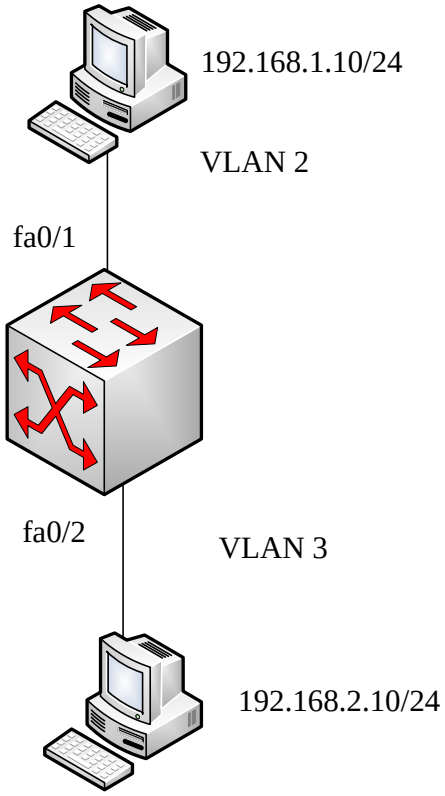


Рисунок 5.7 – Пример сети

Напомним, что у коммутатора третьего уровня имеются логические интерфейсы, привязанные к VLAN. Команды конфигурирования:

Switch(config)#vlan 2

Switch(config-vlan)#name net_2

Switch(config)#vlan 3

Switch(config-vlan)#name net_3

Switch(config)# interface fastEthernet 0/1

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 2

Switch(config)#interface fastEthernet 0/2

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 3

Switch(config)#interface vlan 2

Switch(config-if)#ip address 192.168.1.1 255.255.255.0

Switch(config-if)#no shutdown

Switch(config)#interface vlan 3

Switch(config-if)#ip address 192.168.2.1 255.255.255.0

Switch(config-if)#no shutdown

Как указывалось выше, использование коммутатора третьего уровня лишено недостатков, свойственных использованию маршрутизатора. Однако на практике без маршрутизатора все-таки не обойтись, но используется он, как правило, для связи с внешней сетью. Поэтому классическая схема корпоративной сети с VLAN и с доступом к Интернет имеет вид, показанный на рисунке 5.8.

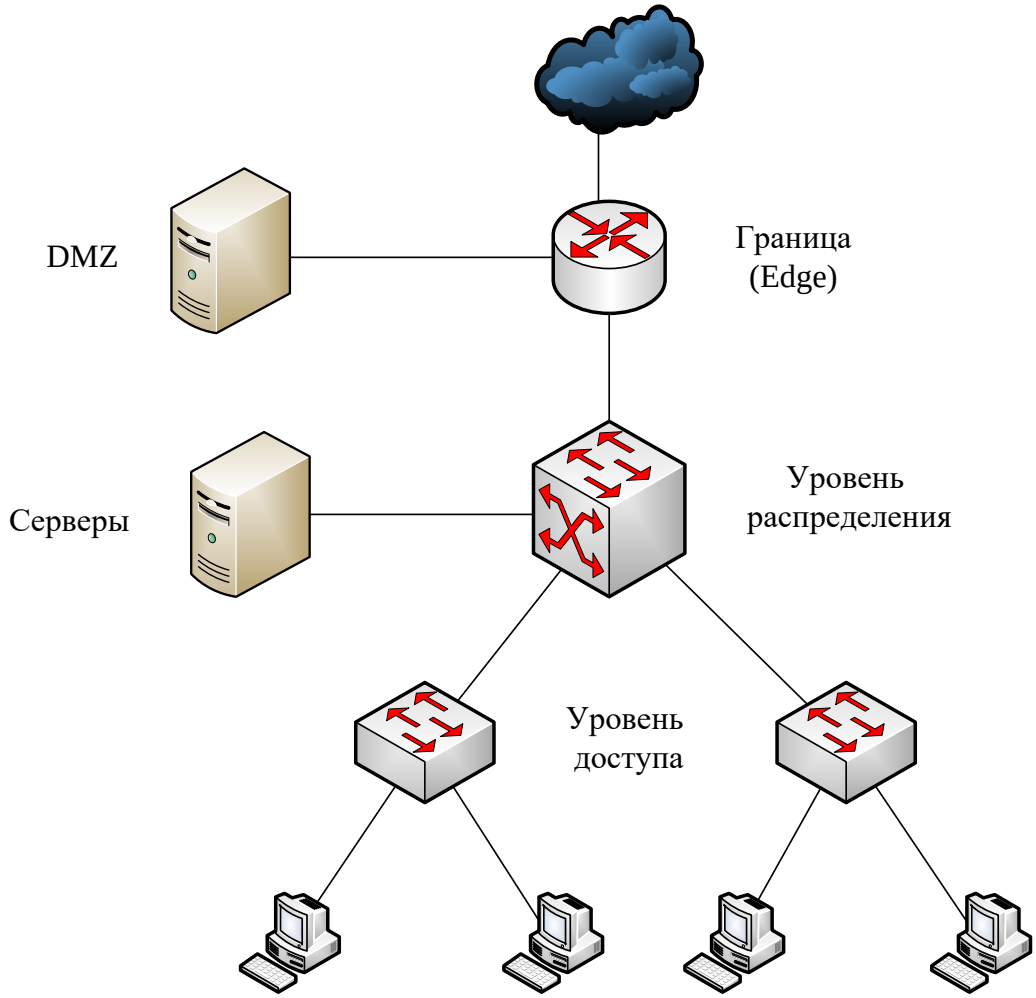


Рисунок 5.8 – Схема корпоративной сети

Рисунок 5.8 достаточно упрощен, в частности, рассматриваемая сеть не является территориально распределенной, однако основные меры защиты, основанные на VLAN на ней рассмотреть можно.

Очевидно, что для разграничения доступа различные категории пользователей должны быть определены в различные VLAN. Внутренние серверы также необходимо выделить в свои VLAN, а при настройке маршрутизации между VLAN должны быть использованы списки доступа. Рассмотрим это на примере.

Сначала необходимо осуществить планирование VLAN. Пример такого планирования представлен в таблице 5.4.

Таблица 5.4 – Планирование VLAN

№ VLAN	Имя	Примечание	Адресация
1	default	Не используется	-
2	management	Управление устройствами	192.168.1.0/24
3	servers	Внутренние серверы	192.168.2.0/24
101	department1	Отдел 1	192.168.3.0/24
102	department2	Отдел 2	192.168.4.0/24
103	department3	Отдел 3	192.168.5.0/24
104	other	Остальные пользователи	192.168.6.0/24

Теперь необходимо составить матрицу доступа (глава 1). Пример такой матрицы представлен в таблице 5.5.

Таблица 5.5 – Матрица доступа

Польз.	Серв. 1	Серв. 2	Серв. 3	Почта	Сет. у-ва
Админ.	SSH	SSH	SSH	SSH	SSH
Отд. 1	All			All	
Отд. 2		All		All	
Отд. 3			All	All	
Остальные				All	

Матрица сильно упрощена, однако достаточна для понимания сути дальнейшего конфигурирования.

Отметим, что серверы выделены в отдельную VLAN 3, однако число серверов – 4, и к ним ограниченный доступ. Поэтому для разграничения доступа к серверам необходимо использовать понятие Private VLAN.

Private VLAN – это виртуальная сеть, включающая в себя вложенные VLAN. При этом вложенная VLAN может быть двух типов:

- Isolated – виртуальная сеть, порты которой не могут обмениваться трафиком между собой;

- Community – виртуальная сеть, порты которой могут обмениваться трафиком между собой, но не могут обмениваться трафиком с портами других Private VLAN.

При этом сеть, включающая в себя вложенные VLAN, называется Primary VLAN, а вложенные VLAN называются Secondary VLAN.

Порт, объединяющий трафик всех вложенных VLAN, называется promiscuous-портом (неразборчивым), или портом, работающим в режиме promiscuous. Порты, через которые проходит трафик своей вложенной VLAN (или одного узла, если вложенный VLAN Isolated), работают в режиме host.

В этом случае серверная VLAN может иметь вид, показанный на рисунке 5.9.

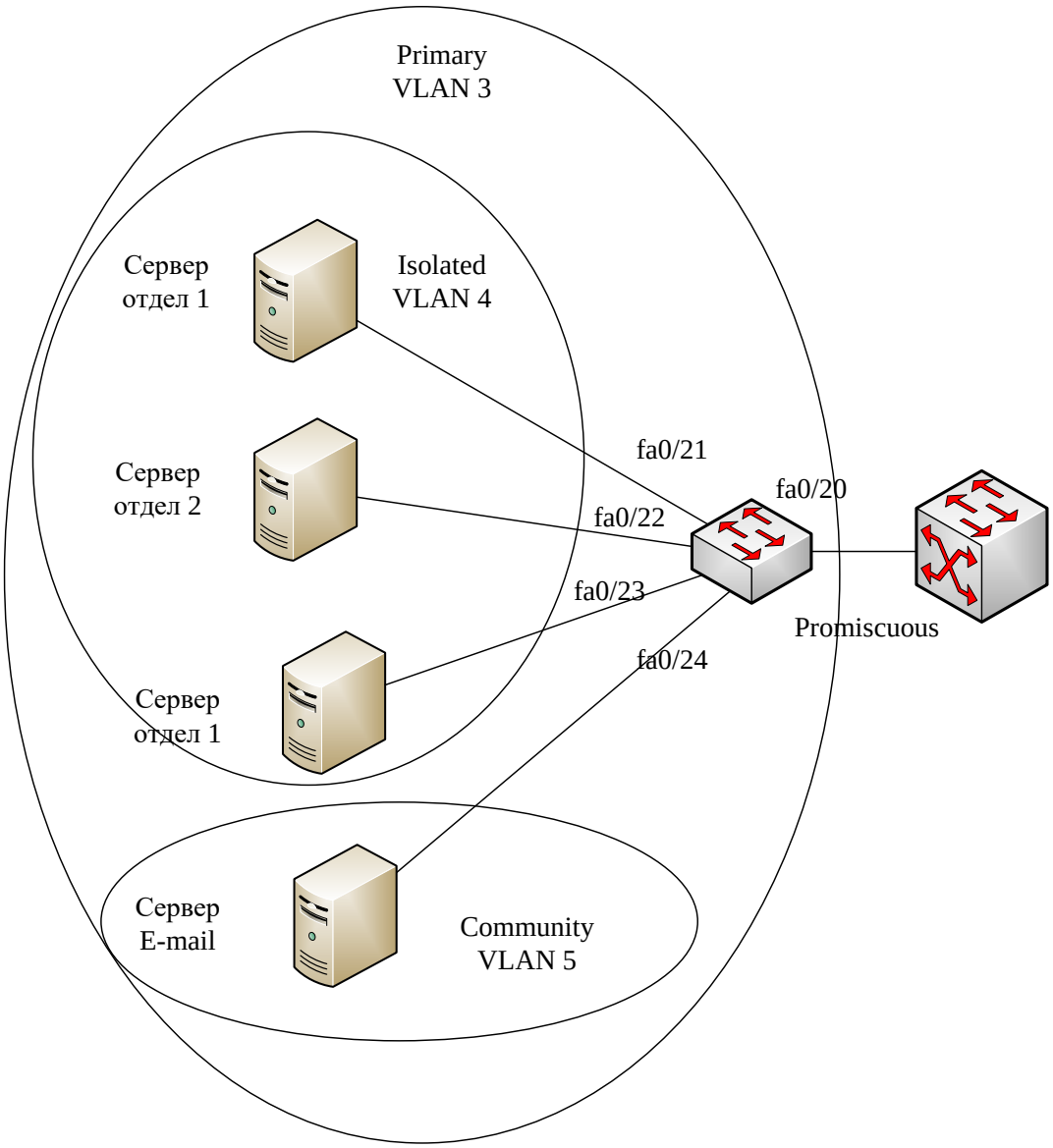


Рисунок 5.9 – Схема серверной VLAN

Рассмотрим процедуру конфигурирования.

Создание VLAN на коммутаторе серверов:

Switch(config)#vlan 2

Switch(config-vlan)#name management

Switch(config)#vlan 101

Switch(config-vlan)#name department1

……. – создание и именование остальных VLAN

Switch(config)#vlan 3

Switch(config-vlan)#name servers

Switch(config-vlan)#private-vlan primary – создание Private VLAN (все серверы)

Switch(config)#vlan 4

Switch(config-vlan)#name departments

Switch(config-vlan)#private-vlan isolated – создание Secondary VLAN (серверы отделов)

Switch(config)#vlan 5

Switch(config-vlan)#name email

Switch(config-vlan)#private-vlan community – создание Secondary VLAN (сервер e-mail)

Switch(config)#vlan 3

Switch(config-vlan)#private-vlan association 4-5 – указание на Secondary VLAN, входящие в Primary VLAN

Привязка физических интерфейсов к VLAN:

Switch(config)#interface fastEthernet 0/x

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan y

Switch(config)#interface fastEthernet 0/21 – интерфейс Сервера отдела 1

Switch(config-if)#switchport mode private-vlan host – перевод порта в режим host

Switch(config-if)#switchport private-vlan host-association 3 4

Switch(config)#interface fastEthernet 0/22 – интерфейс Сервера отдела 2

Switch(config-if)#switchport mode private-vlan host – перевод порта в режим host

Switch(config-if)#switchport private-vlan host-association 3-4

……

Switch(config)#interface fastEthernet 0/20 – порт к коммутатору L3

Switch(config-if)#switchport mode private-vlan promiscuous – перевод порта в режим promiscuous

Switch(config-if)#switchport private-vlan host-association 3 4-5

После этого настраивается маршрутизация между VLAN на коммутаторе третьего уровня с соответствующими списками доступа.

5.3 Защита от атак на VLAN

Рассмотрим сначала наиболее распространенные типы атак на виртуальные частные сети (VLAN). Данные типы атак обычно объединяют термином VLAN Hopping.

Первый тип атак основывается на протоколе DTP (Dynamic Trunking Protocol). Данный протокол позволяет порту коммутатора определить, настроен ли его «сосед» на переход в режим Trunk, и по умолчанию он включен на всех портах коммутаторов Cisco.

Рассмотрим пример соединения двух коммутаторов, рисунок 5.10.

Рисунок 5.10 – Соединение двух коммутаторов

На коммутаторе SW2 порт был переведен в режим Trunk. После подключения его к порту коммутатора SW1 от тоже перейдет в режим Trunk в результате работы протокола DTP. С одной стороны, это удобно, так как нет необходимости настраивать оба коммутатора. С другой стороны, если SW2 является коммутатором нарушителя, он сможет перевести порт коммутатора SW1 в режим Trunk и перехватить трафик всех VLAN.

Рассмотрим более подробно работу протокола DTP. Порты коммутатора могут находиться в одном из режимов:

1. Auto – автоматический режим. DTP-кадры не передаются, но при приеме кадра порт перейдет в режим Trunk.

2. Desirable – режим готовности перейти в состояние Trunk, периодически передает DTP-кадры. При получении ответного DTP-кадра перейдет в режим Trunk.

3. Nonegotiate. В этом режиме отключена рассылка DTP-пакетов с интерфейса. Режим nonegotiate возможен только когда порт находится в режиме access или trunk. Протокол DTP отключен.

4. Trunk.

5. Access.

Особенностью оборудования Cisco является то, что по умолчанию порты находятся в режиме Desirable (в ранних версиях Auto).

Данный тип атаки проиллюстрируем на примере сети, показанной на рисунке 5.11.

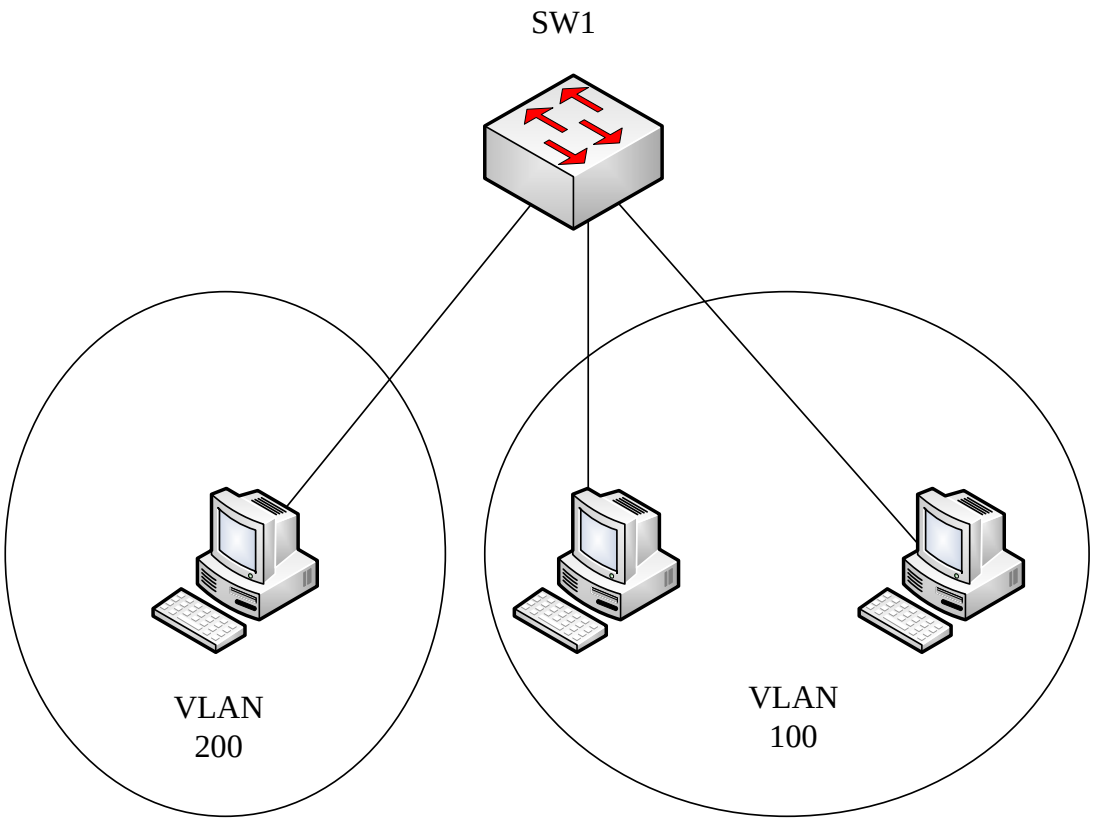


Рисунок 5.11 – Пример сети

Предположим, что правый ПК в VLAN 100 – это компьютер злоумышленника. Он имеет доступ к трафику VLAN 100 и не имеет доступа к трафику VLAN 200. Для перевода порта коммутатора, к которому он подключен, можно использовать ряд утилит, например, yersinia. После этого будет доступен VLAN 200.

Рассмотрим основные способы защиты от атак такого типа.

1. Принудительный перевод портов, к которым подключены ПК, в режим access:

Switch(config)#interface fa0/23

Switch(config-if)#switchport mode access

2. Принудительный перевод портов Trunk в режим nonegotiate:

Switch(config-if)#switchport nonegotiate

3. Разрешение передавать через порт Trunk трафик только определенных VLAN:

Switch(config)#interface fa0/24

Switch(config-if)#switchport mode trunk

Switch(config-if)#switchport trunk allowed vlan 2,3,10-14,20

4. Отключение неиспользуемых портов:

Switch(config)#interface range fa0/21 - 24

Switch(config-if)#shutdown

5. Использование DTP только при начальном конфигурировании сети

Другой тип атак использует VLAN 1 (Native VLAN). Этот VLAN еще называют «VLAN по умолчанию», имея в виду, что весь входящий в порт трафик по умолчанию получает тэг VLAN 1. При передаче, например, по транковому порту этот тэг отбрасывается. Поэтому, если применить двойное тэгирование, где верхний VID=1, после передачи с другого порта этот тэг отбросится, и кадр будет передан с вложенным тэгом. Рассмотрим пример, показанный на рисунке 5.12.

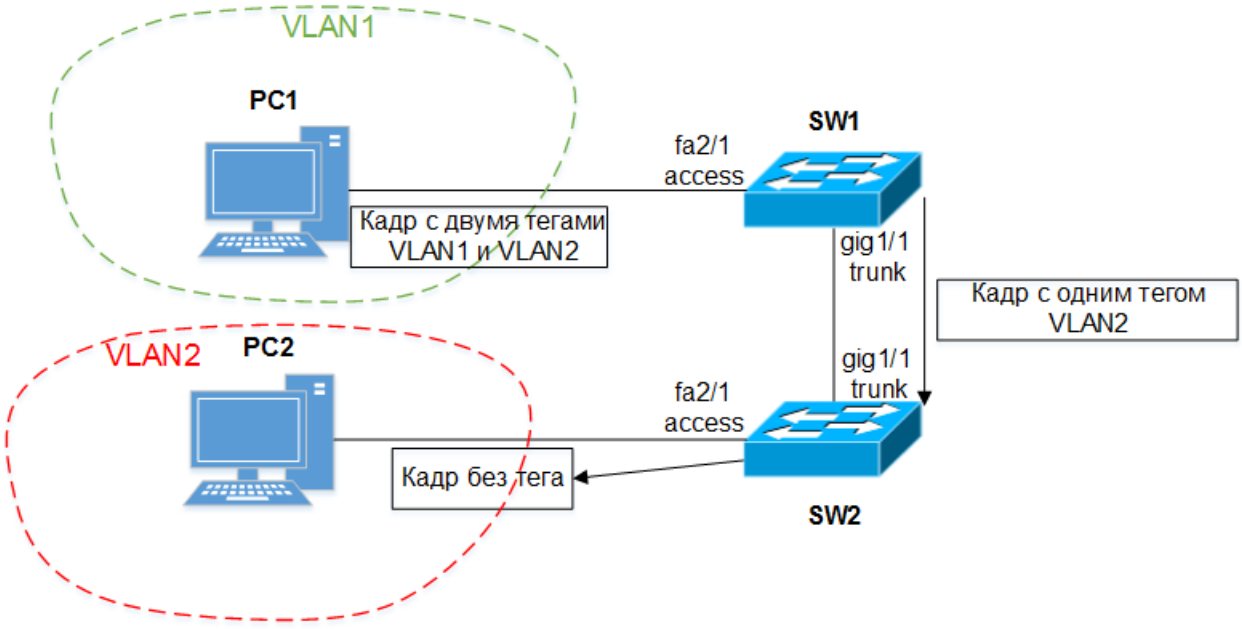


Рисунок 5.12 – Иллюстрация атаки с использованием VLAN 1

В данном примере PC1, являясь компьютером нарушителя, передает на коммутатор SW1 кадр с двумя тэгами, причем верхний тэг имеет VID=1. Коммутатор SW1 должен передать кадр через порт Trunk gig1/1 без тэга, поэтому тэг верхнего уровня будет отброшен. Однако тэг с VID=2 останется,

и коммутатор SW2 воспримет данный кадр как прибывший из VLAN 2. Соответственно, нарушитель попадает в VLAN 2.

Одним из методов защиты от атак этого типа является принудительное тэгирование кадров, принадлежащих к Native VLAN. Тогда при передаче с порта коммутатора кадр будет передаваться не с отброшенным тэгом, а с тэгом той VLAN, которая является Native VLAN (по умолчанию первая). Команда:

Switch(config)#vlan dot1q tag native

Кроме того, как и в случае борьбы с атакой DTP, необходимо разрешить через транковый порт передачу только определенных VLAN, а Native VLAN не использовать вообще.

Наконец, можно в качестве Native VLAN использовать не только VLAN 1, а и любой другой. Поэтому, если указать порту коммутатора использовать по умолчанию VLAN, который нигде в сети не используется, атака неосуществима. Команда:

SW(config-if)#switchport trunk native vlan 999

5.4 Защита от атак на протокол STP

Для начала вспомним принцип работы протокола STP. Данный протокол предназначен для исключения так называемых «петель» в топологии сети.

Под «петлей» понимается такое соединение сетевых устройств между собой, когда между ними существует два или более путей передачи данных.

С одной стороны, «петель» можно избежать при грамотном проектировании сети. Однако при последующей модернизации и расширении сети «петли» могут возникнуть непроизвольно.

С другой стороны, наличие «петель» в сети желательно, так как при существовании нескольких путей передачи данных имеется возможность распараллелить передачу, а также повысить отказоустойчивость сети.

Рассмотрим сначала, к чему в сети на коммутаторах приводит наличие «петли». Рассмотрим для примера сеть, изображенную на рисунке 5.13.

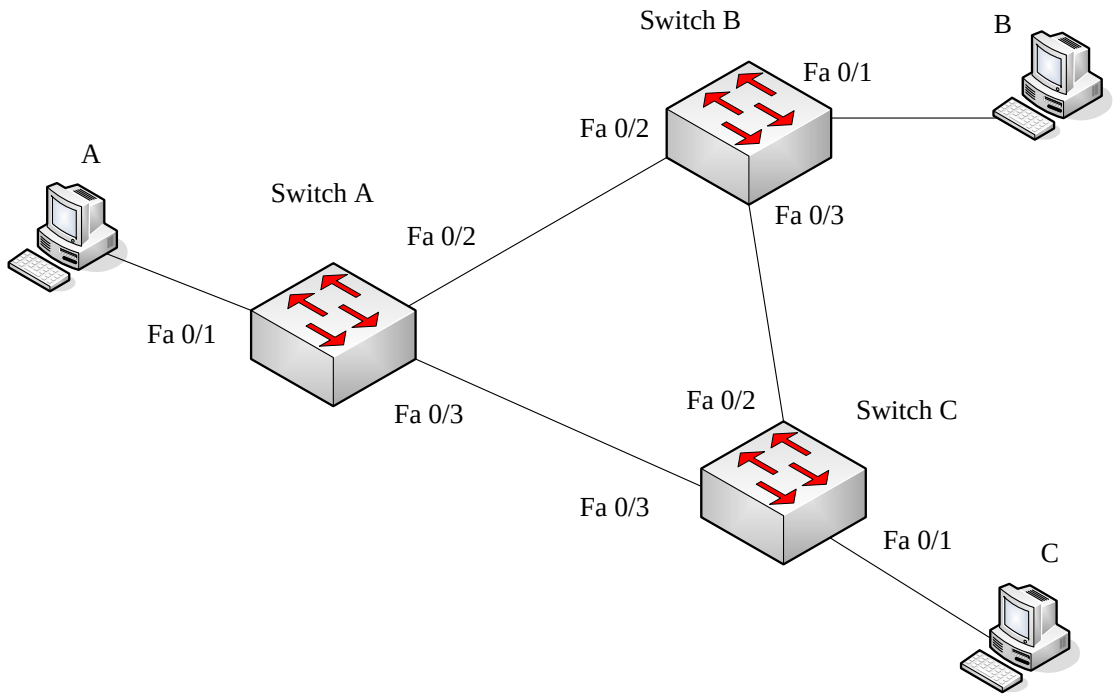


Рисунок 5.13 – Пример сети с «петлей»

Коммутаторы в нашем примере обозначим как Switch A, Switch B, Switch C, а MAC-адреса рабочих станций – A, B, C (однако следует помнить, что реальные MAC-адреса – это шестибайтовые числа). Интерфейсы коммутатора обозначены как Fa 0/1 – Fa 0/3 (Fa – сокращение от Fast Ethernet).

Считаем, что в адресные таблицы коммутаторов не внесены статические записи. Предположим, что в начальный момент времени станция A отправляет широковещательный кадр. Этот кадр поступает на интерфейс Fa 0/1 коммутатора Switch A. Коммутатор ретранслирует этот кадр на порты Fa 0/2 и Fa 0/3, соответственно, его принимают коммутаторы Switch B и Switch C. Одновременно с этим коммутатор Switch A формирует в своей адресной таблице динамическую запись, представленную в таблице 5.6 (для упрощения столбец с указанием номера VLAN в таблице не показан).

Таблица 5.6 – Адресная таблица коммутатора Switch A

Порт	Адрес рабочей станции
Fa 0/1	A

Коммутатор Switch B, приняв кадр, формирует запись, представленную в таблице 5.7.

Таблица 5.7 – Адресная таблица коммутатора Switch B

Порт	Адрес рабочей станции
Fa 0/2	A

Коммутатор Switch C, приняв кадр, формирует запись, представленную в таблице 5.8.

Таблица 5.8 – Адресная таблица коммутатора Switch C

Порт	Адрес рабочей станции
Fa 0/3	A

Так как кадр широковещательный, коммутаторы Switch B и Switch C также распространяют его на все свои задействованные порты. Соответственно, коммутатор Switch B передает его на порт Fa 0/3, в результате чего кадр принимается на порту Fa 0/2 коммутатора Switch C. Коммутатор Switch C корректирует запись в адресной таблице (новая запись представлена в таблице 5.9) и отправляет его на все свои порты, в том числе и на порт Fa 0/3 по направлению к коммутатору Switch A.

Таблица 5.9 – Адресная таблица коммутатора Switch C после корректировки

Порт	Адрес рабочей станции
Fa 0/2	A

Коммутатор Switch A, приняв кадр, тоже корректирует таблицу, как это показано в таблице 5.10.

Таблица 5.10 – Адресная таблица коммутатора Switch A после корректировки

Порт	Адрес рабочей станции
Fa 0/3	A

Аналогичным образом копия того же кадра проходит через коммутатор Switch C в направлении коммутатора Switch B и т.д.

Так как в кадре Ethernet отсутствует поле «время жизни» (Time to Live – TTL), копии одного и того же кадра будут бесконечно циркулировать по «петле» в противоположных направлениях, постоянно корректируя содержимое адресных таблиц всех коммутаторов, через которые они проходят.

Очевидно, что для исключения такой ситуации между любой парой коммутаторов должен существовать единственный путь передачи данных. Даже если сеть небольшая, как в рассмотренном примере, наличие альтернативного пути передачи повысило бы отказоустойчивость сети – при отказе основного пути можно было бы переключиться на резервный.

Для решения этой задачи был разработан алгоритм покрывающего дерева (Spanning Tree Algorithm – STA) и реализующий его протокол покрывающего дерева (Spanning Tree Protocol – STP). Данный протокол описан в открытом стандарте IEEE 802.1D [9]. Реализация данного протокола в современных коммутаторах позволяет отдельные порты перевести в состоянии блокировки (blocking), в котором порты остаются задействованными, но они не передают пользовательский трафик. При отказе неблокированного порта автоматически подключаются заблокированные порты.

Для построения топологии без избыточных каналов строится дерево (математический граф). Чтобы построить такое дерево, вначале необходимо определить корень дерева, из которого и будет строиться граф. Поэтому

первым шагом протокола STP является определение корневого коммутатора (Root Switch). Для определения корневого коммутатора, коммутаторы обмениваются сообщениями BPDU.

Для определения корневого коммутатора используется идентификатор коммутатора — Bridge ID. Bridge ID – это число длиной 8 байт, которое состоит из Bridge Priority (приоритет, 2 байта, от 0 до 65535, по умолчанию 32768) и MAC-адреса устройства. Корневым коммутатором выбирается коммутатор с самым низким приоритетом, если приоритеты равны, то сравниваются MAC-адреса (коммутатор, у которого MAC-адрес меньше, побеждает).

Данную информацию можно посмотреть с использованием команды **show spanning-tree**, выполняемой из привилегированного режима, рисунок 5.14.

```
Switch#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0030.A31A.BBCD
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0030.A31A.BBCD
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20
```

Рисунок 5.14 – Просмотр Bridge ID

Работу протокола STP будем рассматривать на том же примере, рисунок 5.14.

На первом этапе проводится выбор корневого коммутатора (Root Switch). По умолчанию все коммутаторы считают себя корневыми, поэтому они рассылают свои BPDU, в которых указываются Bridge ID. Если коммутатор принимает BPDU с Bridge ID меньшим (лучшим), чем у него, он перестает считать себя корневым и рассылать свои BPDU, но продолжает их транслировать от других коммутаторов. В нашем примере будем считать, что меньший Bridge ID у коммутатора Switch A. Когда Switch B принимает от

Switch A BPDU, он перестает транслировать свои BPDU, то же самое происходит и с коммутатором Switch C. Коммутатор Switch A в результате первого этапа будет избран корневым.

На втором этапе в каждом коммутаторе, кроме корневого, выбирается корневой порт (Root Port). Корневым называется порт, который имеет кратчайшее расстояние до корневого коммутатора.

В качестве расстояния в алгоритме STA используется значение метрики, определяемой как величина, обратно пропорциональная пропускной способности сегмента. В последней версии стандарта IEEE 802.1D 2004 года установлены следующие значения метрик:

- 10 Мбит/с – 2 000 000;
- 100 Мбит/с – 200 000;
- 1 Гбит/с – 20 000;
- 10 Гбит/с – 2 000;
- 100 Гбит/с – 200;
- 10 Тбит/с – 2.

Для определения корневого порта каждый коммутатор использует BPDU, посылаемое корневым коммутатором. При приеме этого сообщения коммутатор определяет значение метрики для конкретного порта, а затем ретранслирует BPDU, предварительно увеличив указанную в нем метрику на значение метрики того сегмента, из которого сообщение было получено. Таким образом, на каждый из портов каждого коммутатора приходят BPDU с различными значениями метрик. Коммутатор выбирает в качестве корневого порта тот порт, для которого метрика имеет минимальное значение.

В нашем примере коммутатор Switch B примет на порту fa0/2 BPDU с метрикой 200 000, а на порту fa0/3 – с метрикой 400 000, так как на этот порт его ретранслирует коммутатор Switch C. Следовательно, порт fa0/2 будет выбран корневым. Аналогично, коммутатор Switch C выберет в качестве корневого порт fa0/3.

В случае, если какой-либо коммутатор принимает на два или несколько портов BPDU с одинаковыми метриками, в качестве корневого выбирается порт с наименьшим идентификатором.

После выбора корневых портов коммутаторы продолжают ретранслировать BPDU, полученные ими от корневого коммутатора, в результате чего они узнают о «петле». Например, коммутатор Switch B, получив от корневого коммутатора BPDU, ретранслирует его на порт fa0/3. Однако на этот же порт приходит BPDU от корневого коммутатора с большей метрикой, следовательно, образовалась «петля».

На третьем этапе реализации алгоритма STA выбирается назначенный коммутатор для каждого сегмента сети и назначенные порты для каждого назначенного коммутатора.

Назначенным называется коммутатор, который в данном сегменте имеет меньшее расстояние до корневого моста. Порт коммутатора, подключенный к назначенному сегменту, называется назначенным портом (Designated Port). У корневого коммутатора все порты назначенные.

После реализации описанных трех этапов коммутаторы блокируют все свои порты, кроме корневых и назначенных. В нашем примере блокироваться может, например, порт fa0/2 коммутатора Switch C.

После окончательной настройки конфигурации определенной топологии коммутаторы продолжают отправлять фреймы BPDU через установленные интервалы. Такие регулярные сообщения BPDU позволяют коммутаторам реагировать на изменения топологии в случае их возникновения.

Перейдем теперь к рассмотрению возможных атак на протокол STP.

Предположим, в сети имеется коммутатор распределения и коммутаторы доступа, как на рисунке 5.8. Коммутатор распределения является корневым. Если нарушитель подключится к двум коммутаторам доступа и начнет рассылать BPDU с нулевым значением Bridge Priority,

протокол STP выберет его в качестве корневого коммутатора. В результате весь трафик будет перехвачен нарушителем, рисунок 5.15.

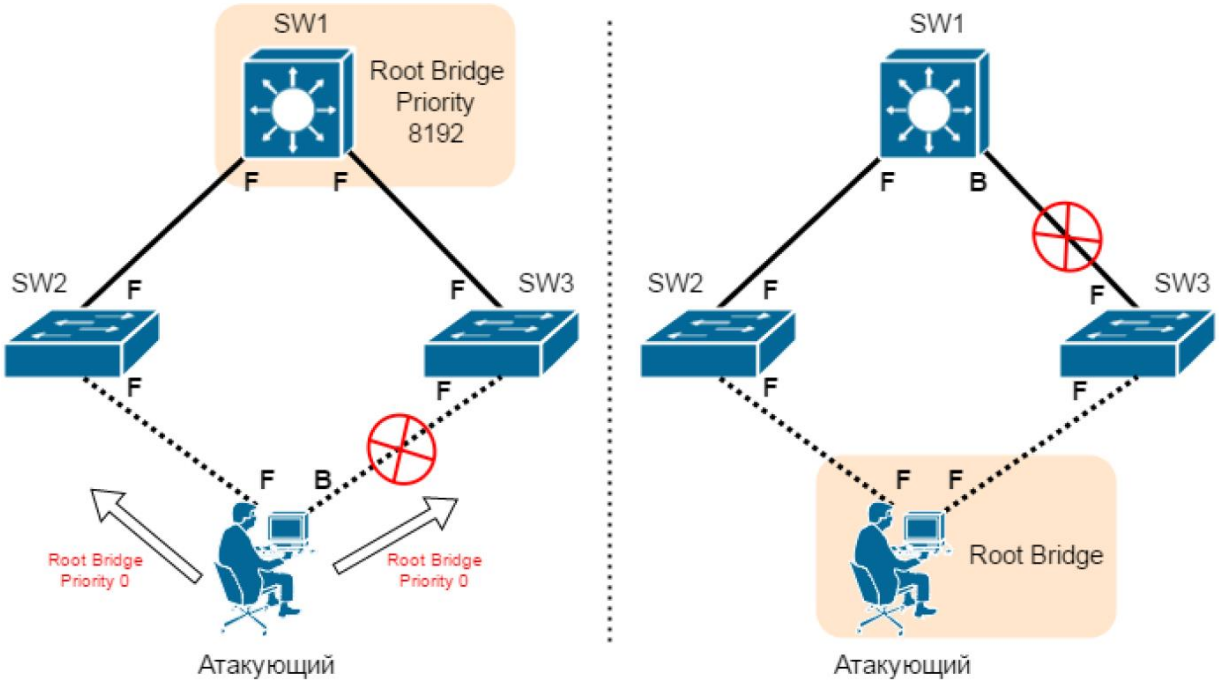


Рисунок 5.15 – Атака на протокол STP

Защита от такой атаки достаточна проста – включение функции BPDU Guard на всех пользовательских портах. Данная функция блокирует передачу через порт кадров BPDU. Команда:

Switch(config-if)spanning-tree bpduguard enable

Однако атакующему не обязательно подключаться к двум коммутаторам. Например, если он подключится только к одному коммутатору и сумеет войти в режим конфигурирования, то появится возможность изменения Bridge Priority. В результате коммутатор, к которому подключился злоумышленник, становится корневым, что позволяет перехватить весь трафик. Кроме того, включая и выключая данный коммутатор, можно полностью вывести сеть из строя, постоянно запуская процесс выборов корневого коммутатора.

В качестве защиты может быть использована функция Root Guard, которая активируется на определенном порту и запрещает появление нового корневого коммутатора. Команда:

Switch1(config-if-range)#spanning-tree guard root

Кроме того, в коммутаторах Cisco можно напрямую указать тот, который будет являться корневым. В этом случае, если будет обнаружено устройство с более низким Bridge ID, будет осуществлен автоматический перерасчет. Например, команда

Switch1(config)#spanning-tree vlan 2,3 root primary

устанавливает, что Switch1 будет являться корневым для VLAN 2 и 3.

5.5 Использование функции Port Security

Технология Port Security предназначена для контроля подключенных к коммутатору устройств и предотвращения аномалий или атак, нацеленных на переполнения таблицы MAC-адресов (CAM table overflow, MAC-flooding).

Например, у коммутатора Cisco 2960 таблица коммутации может содержать до 8024 записей [8]. Поэтому, если подключиться к коммутатору и с помощью специального программного обеспечения сгенерировать большое количество кадров с различными Source Address, таблица переполняется вплоть до стирания всех легитимных MAC-адресов. После этого коммутатор фактически превращается в концентратор, пересылая полученный трафик на все порты, делая его доступным злоумышленнику.

С помощью Port Security устанавливается максимальное количество MAC-адресов на конкретный порт коммутатора или VLAN, и контролируется доступ по заданным MAC-адресам.

Включается функция в режиме конфигурирования интерфейса командой:

Switch(config-if)#switchport port-security

По умолчанию функция ограничивает количество MAC-адресов на интерфейсе одним. Однако на практике к порту может быть подключено

несколько устройств, например, в случае использования IP-телефонии. Команда

Switch(config-if)#switchport port-security maximum <N>

позволяет изменить количество MAC-адресов.

Технология предполагает три режима работы с MAC-адресами:

- Dynamic – запоминание MAC-адресов до тех пор, пока их количество не достигнет разрешенного N;

- Static – указание MAC-адресов вручную;

- Sticky – запоминание MAC-адресов до тех пор, пока их количество не достигнет разрешенного N с записью их в файл конфигурации.

Технология Port Security считает нарушением безопасности следующие ситуации:

- максимальное количество безопасных MAC-адресов было добавлено в таблицу адресов и хост, чей MAC-адрес не записан в таблице адресов, пытается получить доступ через интерфейс;

- адрес, выученный или настроенный как безопасный на одном интерфейсе, появился на другом безопасном интерфейсе в той же VLAN.

В случае, если нарушение произошло, на интерфейсе могут быть настроены следующие действия:

- protect – кадры с неразрешенным MAC-адресом отбрасываются, оповещения о нарушении не выдаются;

- restrict – кадры с неразрешенным MAC-адресом отбрасываются, отправляется оповещение SNMP и SYSLOG, увеличивается счетчик нарушений;

- shutdown – в случае нарушения порт выключается, отправляется оповещение SNMP и SYSLOG, увеличивается счетчик нарушений.

Рассмотрим основные команды конфигурирования.

Switch(config-if)#switchport port-security mac-address 0001.7D16.FF53 – статическая привязка к интерфейсу MAC-адреса;

Switch(config-if)#switchport port-security mac-address sticky – режим Sticky;

Switch(config-if)#switchport port-security violation protect/restrict/shutdown – настройка действия при нарушении;

Функция Port Security может быть настроена и для Trunk-портов, но в этом случае ограничения накладываются на VLAN.

Например, команды:

switch(config)# interface Fastethernet0/3
switch(config-if)#switchport mode trunk
switch(config-if)#switchport port-security maximum 20 vlan 7
switch(config-if)#switchport port-security

переводят интерфейс в режим Trunk и разрешают на нем 20 MAC-адресов в VLAN 7. При этом, если на порту произойдет нарушение, то блокируется трафик только этой VLAN.

5.6 Защита от атак на протоколы DHCP и ARP

Напомним основные принципы работы протокола DHCP.

DHCP является клиент-серверным протоколом, при этом в качестве клиента выступает конечный узел, а в качестве сервера – DHCP-сервер. В соответствии с моделью взаимодействия клиент-сервер конечный узел запрашивает требуемые настройки, а сервер возвращает результат запроса. DHCP-сервер можно организовать на базе любого компьютера сети, а также на базе маршрутизатора. На практике реализация протокола DHCP не требует от сервера высокой производительности, поэтому DHCP-сервер обычно организуется либо на базе имеющегося в сети сервера приложений, либо на базе маршрутизатора.

Рассмотрим работу протокола на примере сети, представленной на рисунке 5.16.

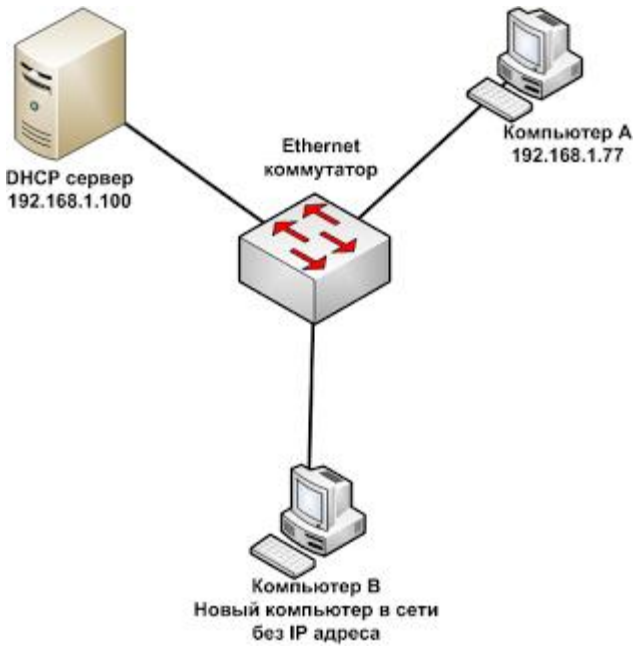


Рисунок 5.16 – Пример сети с DHCP-сервером

Как видно из рисунка, в данной сети имеется DHCP-сервер, коммутатор второго уровня и компьютер, который должен получить от сервера сетевые настройки. Если компьютер работает, например, на базе ОС Windows, для автоматической настройки сетевых параметров с использованием протокола DHCP необходимо под правами администратора зайти на вкладку «Свойства используемого подключения», дважды щелкнуть на пункте «Протокол Интернета (TCP/IP)», и в отобразившемся окне выбрать пункт «Получить IP-адрес автоматически» (рисунок 5.17).

В этом случае компьютер выступает в качестве клиента, который будет пытаться получить у сервера сетевые настройки. Для этого он формирует запрос на широковещательный адрес 255.255.255.255, а в качестве своего IP-адреса указывает адрес 0.0.0.0. Данный кадр называется DHCP DISCOVER, в нем указывается MAC-адрес узла, сформировавшего данный запрос.

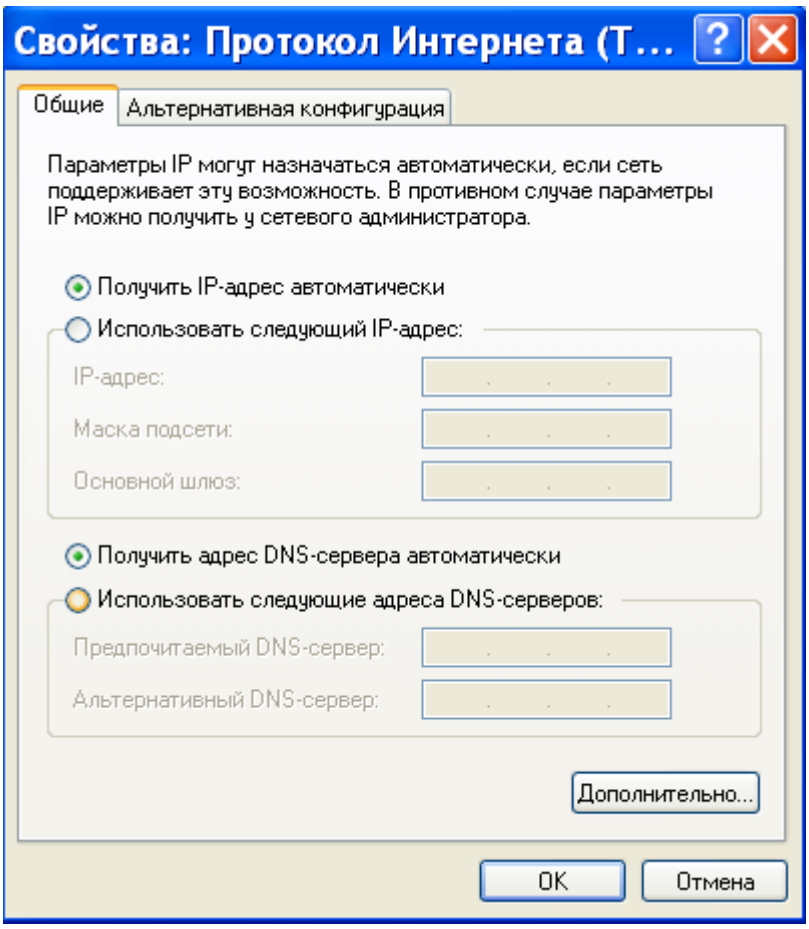


Рисунок 5.17 – Автоматическое получение сетевых настроек

Так как переданный кадр-запрос является широковещательным, его получают все узлы сети, включая и DHCP-сервер, то есть в нашем примере запрос получают как компьютер A с адресом 192.168.1.77, так и сервер с адресом 192.168.1.100. Компьютер A запрос игнорирует, так как он не является DHCP-сервером.

Сервер, приняв запрос, выбирает в соответствии со своими настройками подходящую конфигурацию, и отправляет ее в кадре-ответе, который называется DHCP OFFER. В отличие от DHCP DISCOVER, сообщение DHCP OFFER является адресным, то есть пересылается не широковещательно, а на MAC-адрес узла, приславшего запрос.

Компьютер A, приняв сообщение DHCP OFFER, запоминает содержащиеся в нем сетевые настройки, и отсылает серверу сообщение DHCP REQUEST, имеющее такую же структуру, как и DHCP DISCOVER, но

с указанием IP-адреса DHCP-сервера. Это сообщение также рассылается широковещательно (так как в сети может быть несколько DHCP-серверов).

Сервер, приняв сообщение DHCP REQUEST со своим IP-адресом, подтверждает это ответным сообщением DHCP ACK, и помечает выделенный IP-адрес как задействованный. Рабочая станция, получив сообщение DHCP ACK, применяет сохраненные ранее настройки.

В свою очередь, DHCP-сервер также нуждается в конфигурировании – по меньшей мере, на нем должен быть задан пул адресов, которые он должен выдавать рабочим станциям, и время аренды, на которое выдается IP-адрес. Если DHCP-сервер организуется на базе маршрутизатора, объединяющего несколько подсетей, при конфигурировании протокола DHCP необходимо указать интерфейс, который будет использоваться для раздачи сетевых настроек в пределах подсети.

Рассмотрим пример настройки DHCP-сервера, развернутого на базе маршрутизатора Cisco. Используя Cisco Packet Tracer, построим простейшую сеть, представленную на рисунке 5.18.

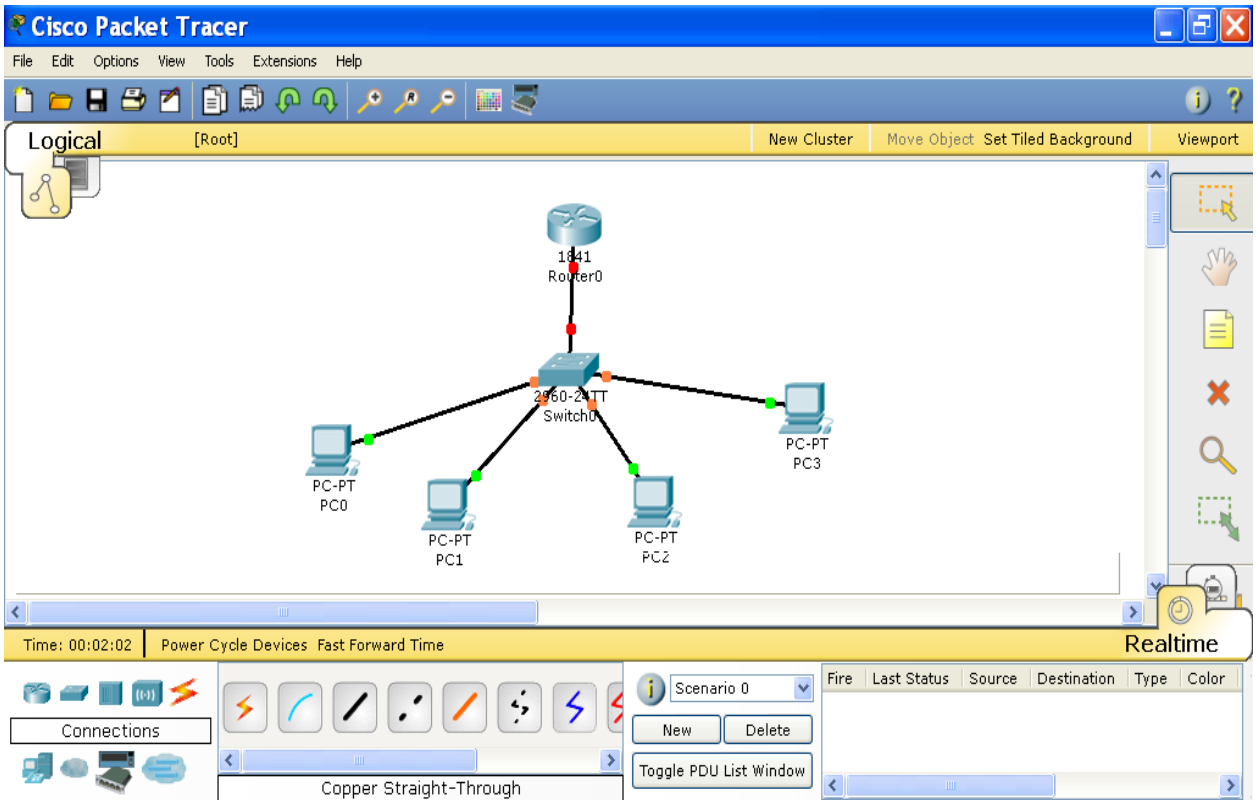


Рисунок 5.18 – Пример сети

На рабочих станциях необходимо выбрать пункт DHCP на вкладке IP Configurations. На реальном компьютере, работающем, например, под управлением ОС Windows, необходимо выбрать режим автоматического получения настроек, как это показано на рисунке 5.17.

Зададим внутреннему порту маршрутизатора IP-адрес, рисунок 5.19.

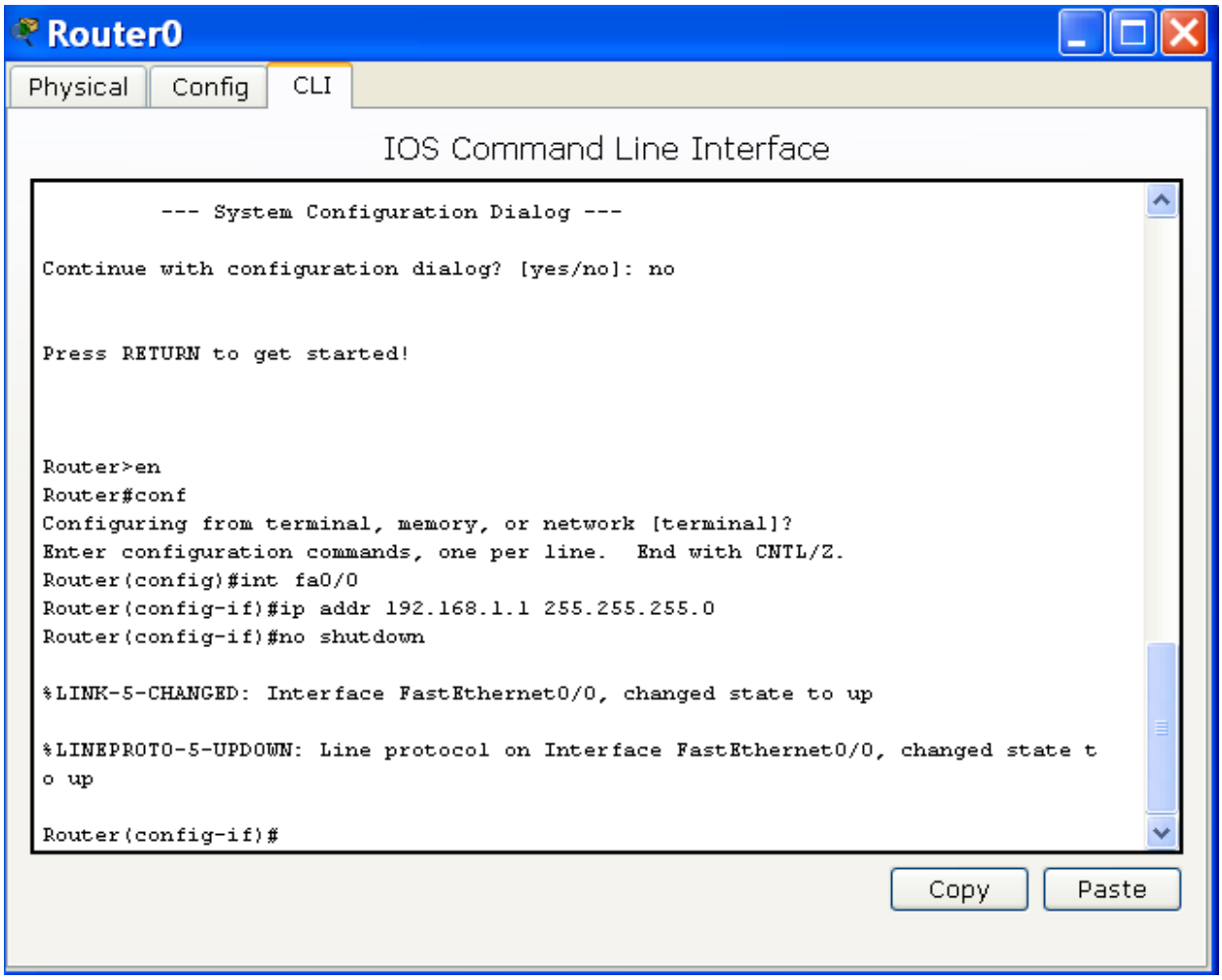


Рисунок 5.19 – Конфигурирование маршрутизатора

По умолчанию на устройствах Cisco функция DHCP-сервера включена. Если же она ранее была выключена, включить ее можно командой

Router(config)#service dhcp,

выполняемой в режиме глобального конфигурирования.

Для конфигурирования DHCP-сервера используются следующие команды:

Router(config)#ip dhcp pool <имя> - создание пула адресов;

Router(dhcp-config)#network <адрес сети> <маска> – объявление адреса сети, из которой будут раздаваться IP-адреса;

Router(dhcp-config)#default-router <адрес> – назначение адреса шлюза по умолчанию;

Router(config)#ip dhcp excluded-addr <адрес> – исключение адреса (или адресов) из пула раздаваемых адресов;

Зайдя на вкладку IP Configurations любого компьютера, убедимся, что компьютер получил от сервера сетевые настройки, рисунок 5.20.

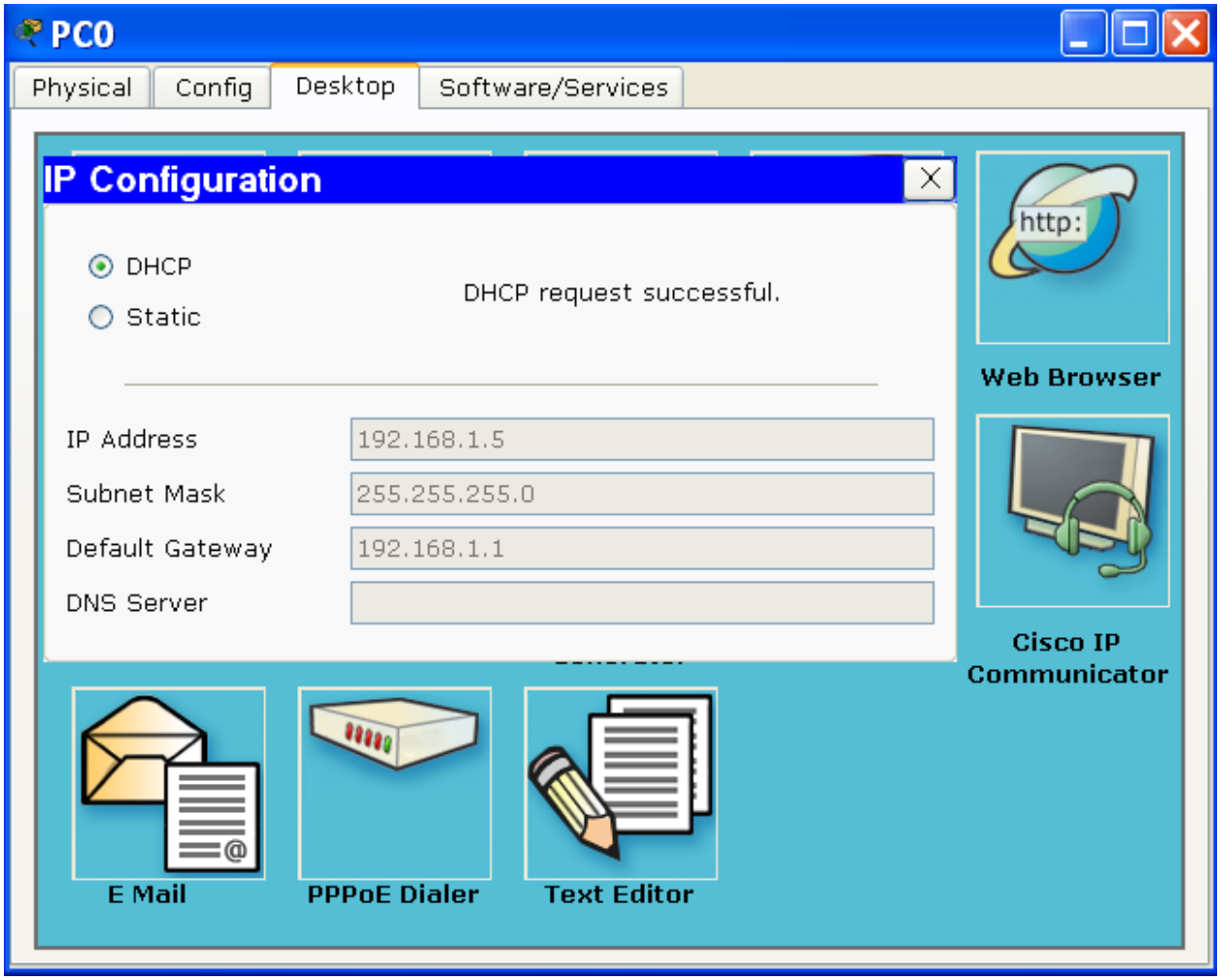


Рисунок 5.20 – Получение рабочей станцией сетевых настроек

Протокол DHCP изначально никаким образом не защищен, в нем нет механизмов проверки подлинности ответов от DHCP-сервера, отсутствует аутентификация клиента, и т.д.

Поэтому с использованием DHCP возможны несколько типов атак.

Во-первых, можно обеспечить отказ в обслуживании (DoS) за счет отсылки на DHCP-сервер множества запросов с различных MAC-адресов. В результате DHCP-сервер выдаст все адреса из настроенного на нем пула, и легальные пользователи не получат сетевых настроек.

Во-вторых, злоумышленник может создать поддельный DHCP-сервер. Так как рассылка запроса DHCP Discover происходит широковещательно, ее получают все узлы сети. Соответственно, можно выдать себя за сервер, сформировать ответ, в котором в качестве шлюза по умолчанию указать собственный адрес. После принятия рабочими станциями полученных настроек весь трафик будет передаваться через нарушителя, соответственно, будет иметь место атака «человек посередине» (Man In The Middle), рисунок 5.21.

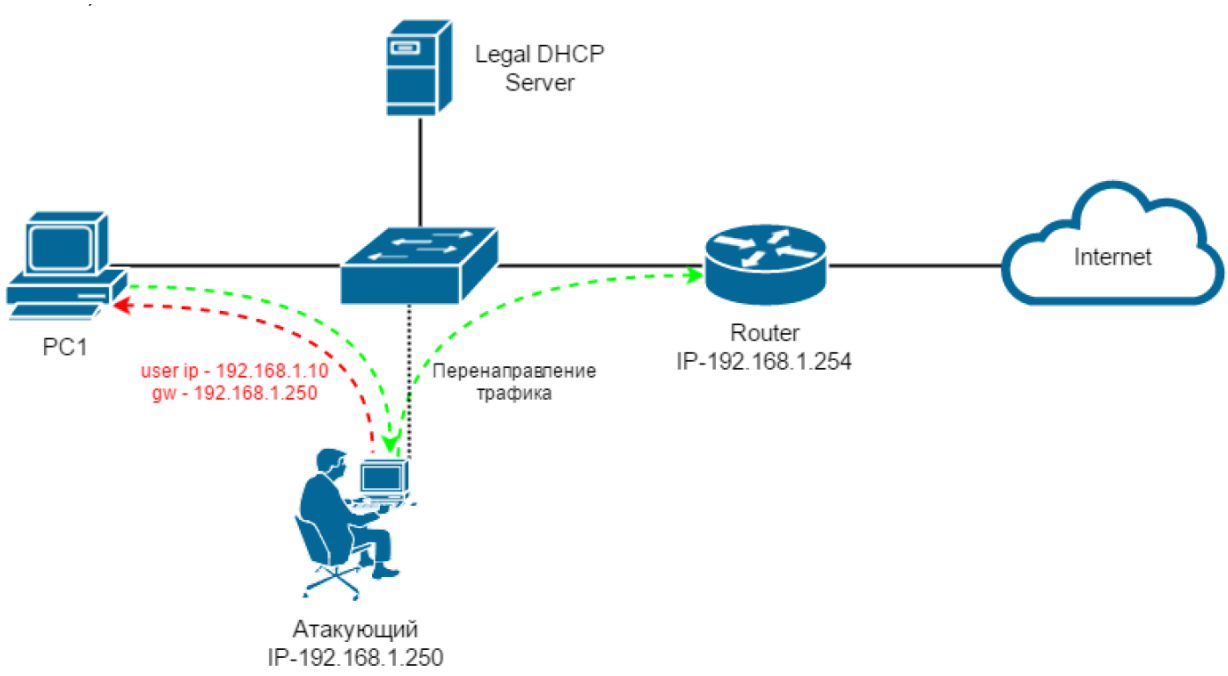


Рисунок 5.21 – Атака «человек посередине»

Для защиты от таких атак используется функция DHCP snooping. Основная идея заключается в том, что все порты разделяются на доверенные (trust) и недоверенные (untrust). DHCP-сервер должен располагаться только

за доверенными портами. Ответы DHCP-сервера, пришедшие на недоверенные порты, будут отбрасываться. Команды:

Switch(config)#ip dhcp snooping

Switch(config)#ip dhcp snooping vlan 2

Switch(config)#int fa0/15

Switch(config-if)#ip dhcp snooping trust

Первой командой на коммутаторе глобально включается функция DHCP snooping. Вторая команда определяет, для каких VLAN включается защита (так как широковещательный кадр распространяется в приделах VLAN). Третьей командой осуществляется вход в режим конфигурирования порта fa0/15, четвертая команда объявляет его доверенным портом. После этого остальные порты будут считаться недоверенными.

Очевидно, что доверенным должен быть объявлен порт, за которым располагается легальный DHCP-сервер, а все остальные порты, в особенности те, к которым подключены пользователи, должны быть недоверенными.

Функция DHCP snooping выполняет еще одну важную задачу. При включении этой функции коммутатор автоматически создаст базу соответствия портов, MAC и IP-адресов. Эта база формируется за счет отслеживания сообщений протокола DHCP, которые проходят через коммутатор, и хранится в таблице соответствия (Binding Table). Это дает возможность борьбы с атаками подмены IP-адреса отправителя (IP Spoofing). Такая защитная функция получила название IP Source Guard. Таблицу соответствия рекомендуется создавать для недоверенных портов, к которым подключены рабочие станции. Таблица соответствия, созданная для недоверенного Trunk-порта, может привести к переполнению памяти коммутатора.

Рассмотрим эту функцию подробнее на примере схемы, показанной на рисунке 5.22.

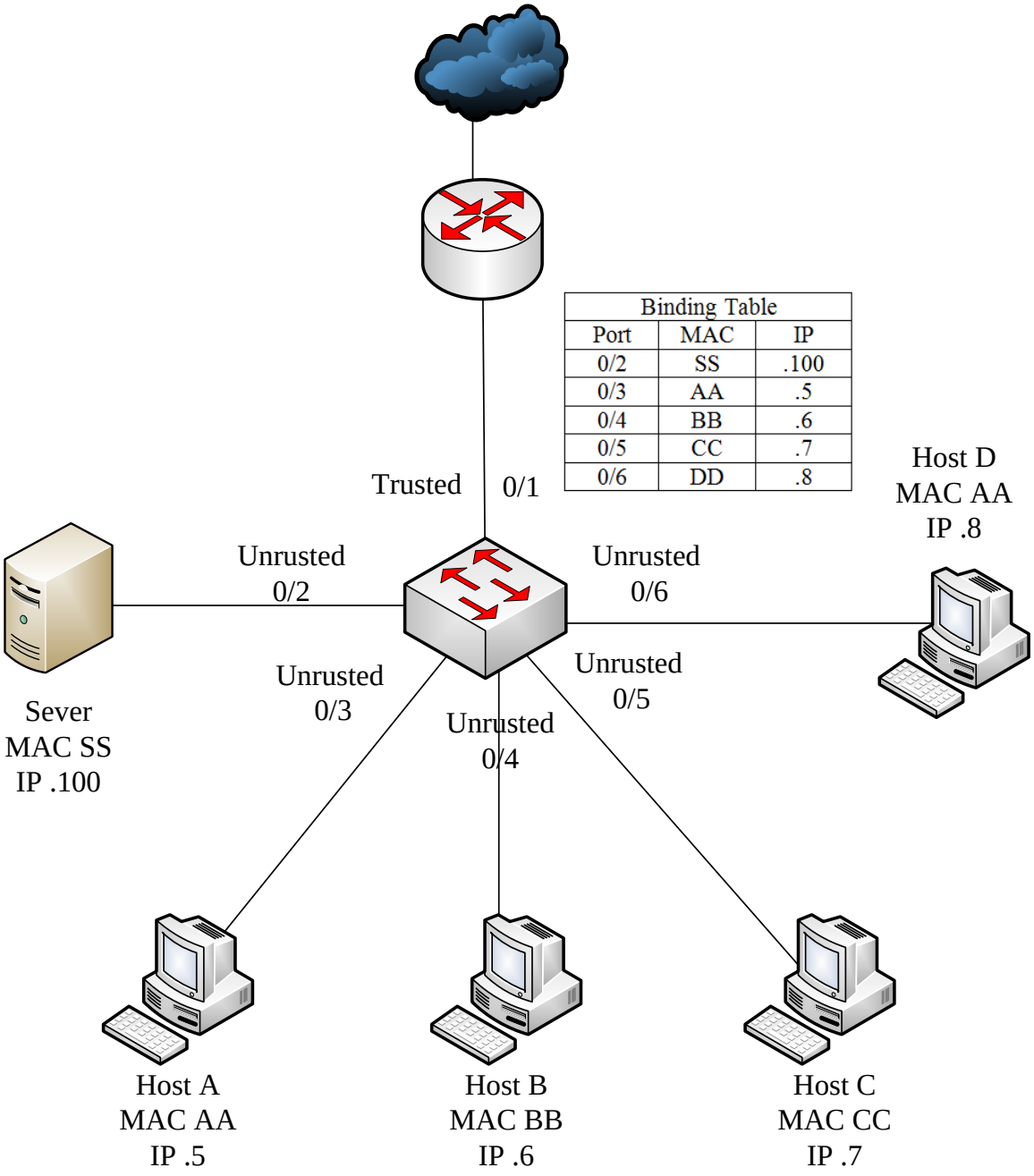


Рисунок 5.22 – Иллюстрация работы Source Guard

Как видим, в таблице соответствия находятся актуальные записи. Если злоумышленник, находящийся, например, за хостом D, пошлет на порт коммутатора 0/6 пакет с адресом источника, отличающимся от .8, будет обнаружено несоответствие, и пакет будет отброшен.

Включается функция командой для конкретного интерфейса:

Switch(config)#int fa0/3

Switch(config-if)#ip verify source vlan dhcp-snooping port-security

Перейдем теперь к рассмотрению протокола ARP.

Как известно, по мере продвижения пакета по составной сети адреса канального уровня (локальные адреса) изменяются от одной подсети к другой, адреса сетевого уровня (IP-адреса) остаются неизменными. Маршрутизатор, декапсулировав пакет из принятого кадра, определяет по таблице маршрутизации интерфейс, на который его нужно продвинуть. На этом интерфейсе пакет инкапсулируется в кадр канальной технологии (например, Ethernet) и передается по подсети следующему маршрутизатору или конечному узлу. Таким образом, для передачи пакета через подсеть необходимо знать локальный адрес порта следующего маршрутизатора или конечного узла. Иначе говоря, по IP-адресу порта следующего маршрутизатора или конечного узла необходимо определить его локальный адрес (MAC-адрес, если подсеть использует Ethernet).

Эта задача решается с использованием протокола ARP (Address Resolution Protocol – протокол разрешения адреса).

В соответствии с данным протоколом все сетевые устройства содержат в своей памяти ARP-таблицу, связывающую сетевые и локальные адреса. Просмотреть содержимое этой таблицы на конечном узле, работающем под управлением ОС Windows, можно с использованием команды arp –a, рисунок 5.23.

Из рисунка 5.23 следует, что IP-адресу 192.168.1.1 соответствует MAC-адрес 14:da:e9:5a:6b:68, в данном примере это адрес шлюза по умолчанию. Так как это таблица конечного узла, она содержит только одну запись, соответствующую маршруту по умолчанию. У маршрутизаторов таких записей несколько, однако их ARP-таблицы не могут содержать всех необходимых записей – их оказалось бы недопустимо много.

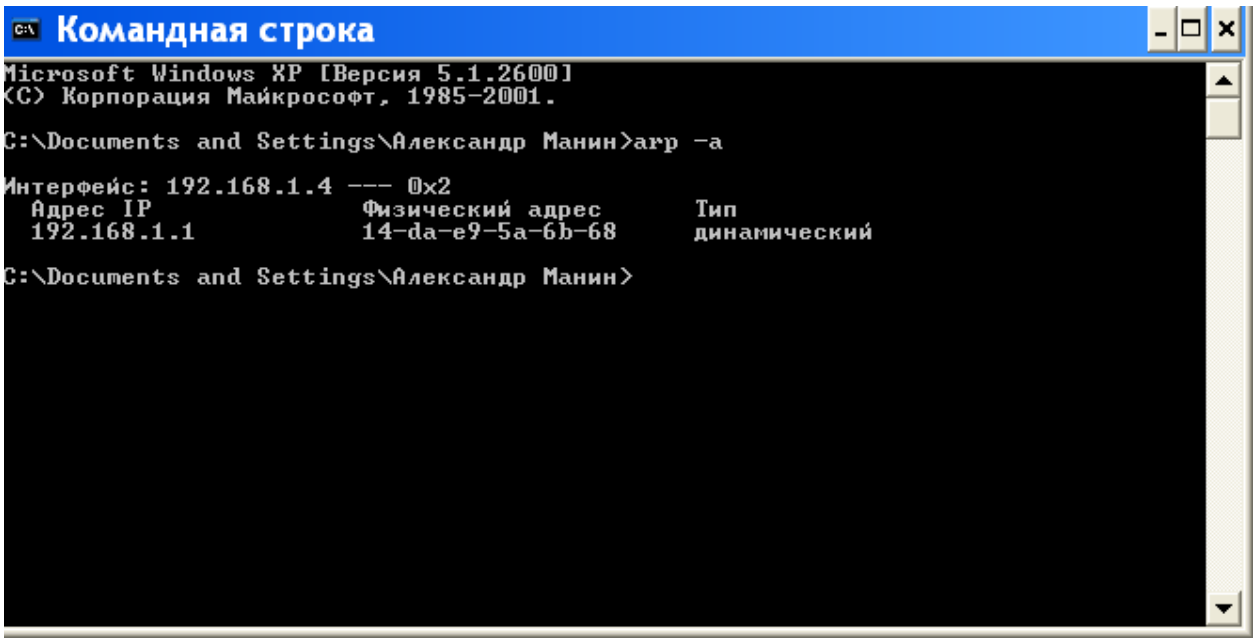


Рисунок 5.23 – Просмотр ARP-таблицы в ОС Windows

Поэтому при необходимости передачи пакета через подсеть протокол IP сначала обращается к ARP-таблице и пытается найти там нужный локальный адрес по заданному IP-адресу. Если такой записи в таблице нет, протокол ARP формирует ARP-запрос, инкапсулирует его в кадр канальной технологии (например, Ethernet), и широковещательно передает в пределах подсети. Запрос содержит IP-адрес порта назначения. Приняв запрос, устройство, чей IP-адрес совпадет с IP-адресом, содержащимся в запросе, формирует ARP-ответ, содержащий локальный адрес. Полученная в результате такого обмена информацией пара IP-адрес – локальный адрес записывается в ARP-таблицу (кэшируется), так как велика вероятность передачи следующего пакета по этому же адресу.

Протокол ARP также является абсолютно незащищенным. Когда в сеть отправляется широковещательный ARP-запрос, ответить на него может абсолютно любое устройство, в том числе и компьютер злоумышленника, который в итоге получает возможность перехвата трафика. Такую атаку часто называют ARP-spoofing.

Для предотвращения таких атак используется функция Dynamic ARP Inspection (DAI). Принцип работы аналогичен DHCP snooping – порты

подразделяются на доверенные и недоверенные. На недоверенных портах ARP-ответы подвергаются анализу. Например, если включена функция Port Security, можно проверить MAC-адрес, при включенном IP Source Guard можно проверить соответствие MAC и IP-адресов.

Для активации DAI используется команда:

Switch(config)#ip arp inspection vlan <№>

После активации в данной VLAN будет разрешен трафик только тех устройств, данные о которых имеются в таблице соответствия DHCP snooping.

Для объявления порта доверенным используется команда:

Switch(config-if)#ip arp inspection trust

Еще раз отметим, что здесь рассмотрены лишь базовые приемы защиты.

5.7 Практическое задание

5.7.1 Собрать в Cisco Packet Tracer схему сети, показанную на рисунке 5.24. Настроить адресацию, обеспечить доступность внешнего сервера м всех внутренних серверов с внутренних ПК. Обеспечить выдачу ПК динамических адресов с использованием DHCP-сервера, развернутого на коммутаторе третьего уровня. Адрес внешнего сервера выбрать произвольно из адресного пространства общедоступных адресов. Адреса внутренних адресов выбрать произвольно.

ПРИМЕЧАНИЕ. Для перевода порта коммутатора третьего уровня в режим Trunk необходимо сначала включить на нем режим desirable:

Switch(config-if)#switchport mode dynamic desirable

5.7.2 Сконфигурировать на маршрутизаторе динамический NAT с перенаправлением портов. Убедиться в работоспособности, просмотреть статистику трансляций.

5.7.3 С использованием ACL обеспечить доступность пользователей VLAN 101 к FTP-серверу VLAN 6, пользователей VLAN 102 – к FTP-серверу VLAN 7. Доступ к внешнему серверу предоставить всем пользователям.

ПРИМЕЧАНИЕ. Если технология Private VLAN не поддерживается используемой версией Cisco Packet Tracer, разместить серверы в обычных VLAN.

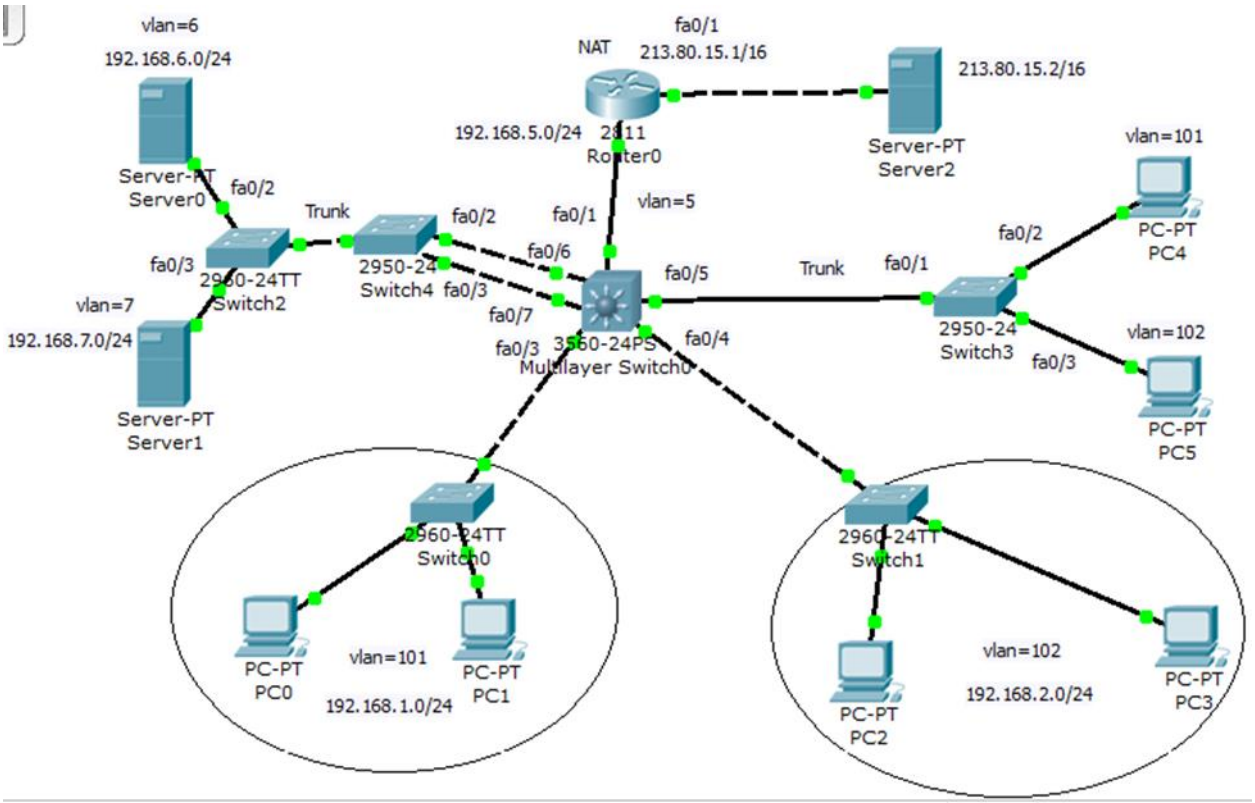


Рисунок 5.24 – Схема сети

5.7.4 С использованием функции Port Security привязать к пользовательским портам коммутаторов доступа MAC-адреса подключенных ПК. Подключив к этим портам другие ПК, убедиться в работоспособности системы защиты.

ПРИМЕЧАНИЕ. Для включения функции Port Security порт должен находиться не в динамическом режиме, а в режиме доступа (mode access).

6 Виртуальные частные сети

6.1 Технологии туннелирования. Протокол GRE

Рассмотрим сначала классификацию VPN по назначению, так как именно от этого зависит настройка.

Remote Access VPN – используется для создания защищённого канала между сегментом корпоративной сети (центральным офисом или филиалом) и одиночным пользователем, который, работая дома, подключается к корпоративным ресурсам с домашнего компьютера, корпоративного ноутбука, смартфона. В терминологии Cisco такой пользователь называется Teleworker.

Site-to-Site VPN – используется для создания защищенного канала между различными сегментами корпоративной сети (центральным офисом и филиалами) через незащищенную сеть (Интернет).

Независимо от типа VPN при его конфигурировании создается туннель. Туннелирование – это процесс, в ходе которого создается защищенное логическое соединение между двумя конечными точками посредством инкапсуляции различных протоколов. При туннелировании данные упаковываются вместе со служебными заголовками в новый «конверт» для обеспечения конфиденциальности и целостности всей передаваемой информации.

Понятие «туннелирование» включает в себя ряд терминов:

- транспортируемый протокол (протокол-«пассажир») – протокол, в котором представлены данные, которые необходимо передать через туннель;

- транспортирующий (несущий) протокол – протокол, на базе которого данные доставляются через некоторую транзитную сеть (например, через Интернет);

- протокол инкапсуляции – протокол, описывающие правила инкапсуляции данных транспортируемого протокола в пакет транспортирующего протокола.

Протокол транзитной сети является несущим, а протокол объединяемых сетей — транспортируемым. Пакеты транспортируемого протокола помещаются в поле данных пакетов несущего протокола с помощью протокола инкапсуляции. Пакеты-«пассажиры» не обрабатываются при транспортировке по транзитной сети никаким образом (по сути, являются полем данных для транспортирующего протокола). Инкапсуляцию выполняет пограничное устройство (маршрутизатор или шлюз), которое находится на границе между исходной и транзитной сетями. Извлечение пакетов транспортируемого протокола из несущих пакетов выполняет второе пограничное устройство, расположенное на границе между транзитной сетью и сетью назначения. Пограничные устройства указывают в несущих пакетах свои адреса, а не адреса узлов в сети назначения.

Туннель может быть использован не только для создания VPN, но и когда две сети с одной транспортной технологией необходимо соединить через сеть, использующую другую транспортную технологию. При этом пограничные маршрутизаторы, которые подключают объединяемые сети к транзитной, упаковывают пакеты транспортируемого протокола объединяемых сетей в пакеты транспортирующего протокола транзитной сети. Второй пограничный маршрутизатор выполняет обратную операцию. Например, с использованием туннелирования можно передавать трафик протоколов IPX, IPv6, AppleTalk через IPv4-сеть. В этом случае IPX, IPv6, AppleTalk будут являться транспортируемыми протоколами, а IPv4 – транспортирующим протоколом.

Туннелирование может использоваться на разных уровнях модели OSI, например, на канальном, или сетевом. Здесь будем рассматривать только туннелирование на сетевом уровне.

Одним из наиболее распространенных протоколов инкапсуляции, используемом для создания туннелей на сетевом уровне, является протокол GRE – Generic Routing Encapsulations [9]. GRE – это протокол, разработанный компанией Cisco Systems, позволяющий инкапсулировать пакеты разного типа внутри IP-туннелей. Благодаря этому создается виртуальный канал «точка-точка» между маршрутизаторами Cisco поверх IP-сети. Необходимо отметить, что само по себе туннелирование не обеспечивает функционал VPN – данные транспортируемого протокола остаются незашифрованными и легко поддаются перехвату.

Инкапсуляция в соответствии с протоколом GRE в случае, если транспортируемым протоколом является IP, иллюстрируется рисунком 6.1.

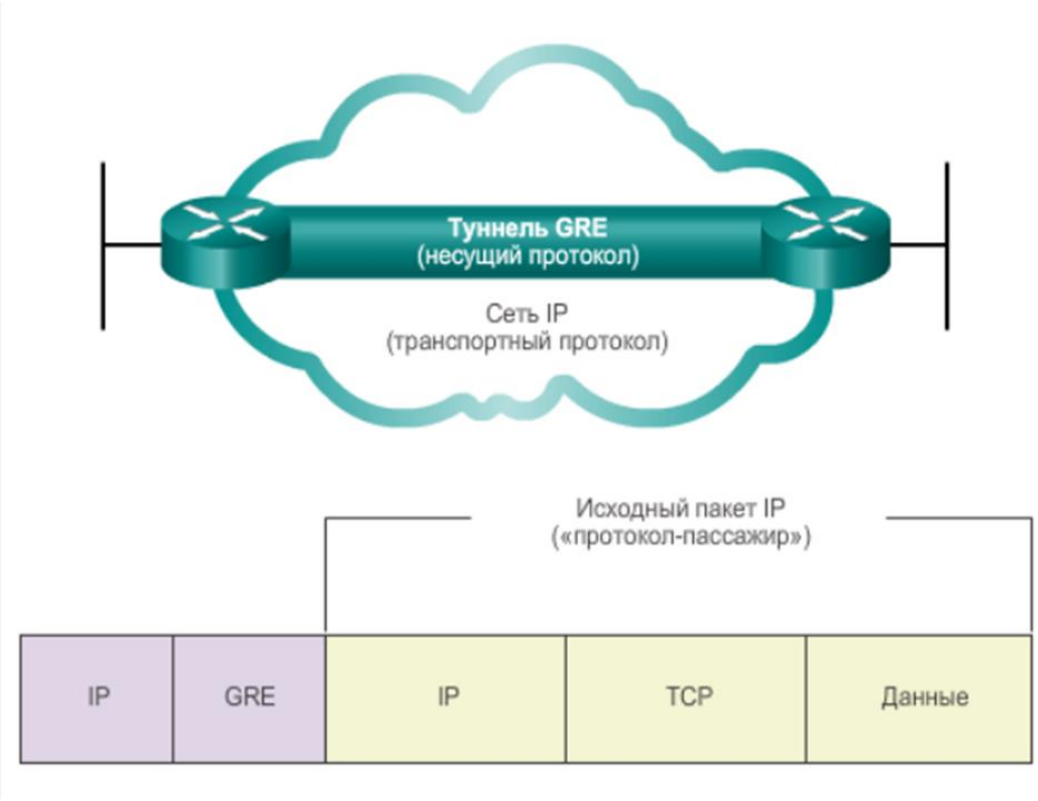


Рисунок 1.1 – Инкапсуляция с использованием GRE

Рассмотрим пример, заимствованный из [10].

Имеется общедоступная сеть, к которой выполнено подсоединение двух локальных сетей LAN1 и LAN2 с использованием маршрутизаторов Router_A и Router_B. Отметим, что в локальных сетях может использоваться

адресация из диапазона частных немаршрутизируемых адресов, например, сеть, подключенная к Router_A, имеет адрес 192.168.1.0/24, а сеть, подключенная к Router_B – адрес 192.168.3.0/24, рисунок 6.2.

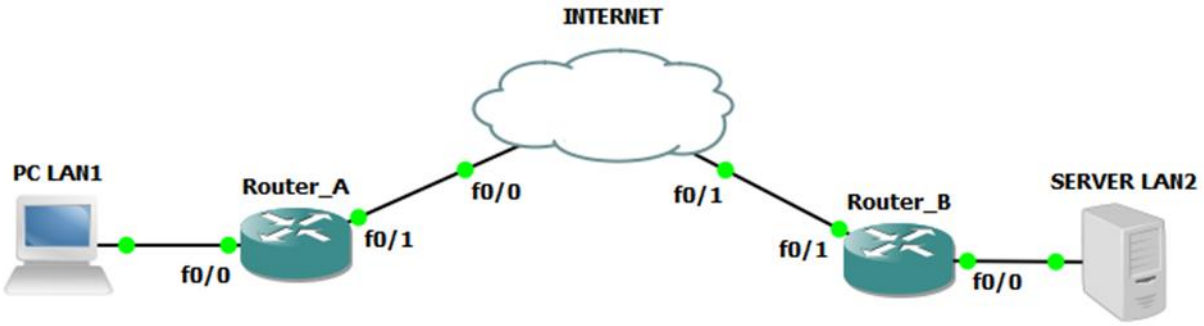


Рисунок 6.2 – Пример объединения двух частных сетей

Адресация в сети следующая:

Router_A:

f0/0 – 192.168.1.1/24

f0/1 – 11.11.11.11/8

Router_B:

f0/0 – 192.168.3.1/24

f0/1 – 33.33.33.33/8

Кроме реальных адресов, в рассматриваемой сети необходима адресация туннеля. Так как туннель представляет собой вырожденную сеть (точка-точка), можно использовать для него адрес 192.168.2.0/30.

Рассмотрим команды конфигурирования маршрутизатора Router_A:

Router_A(config)#interface fastEthernet0/0

Router_A(config-if)#ip address 192.168.1.1 255.255.255.0

Router_A(config-if)#no shutdown

Router_A(config-if)#exit

Router_A(config)#interface fastEthernet0/1

Router_A(config-if)#ip address 11.11.11.11 255.0.0.0

Router_A(config-if)#no shutdown

Router_A(config-if)#exit

Это были обычные команды конфигурирования интерфейсов маршрутизатора. Теперь необходимо сконфигурировать создаваемый туннель:

Router_A(config)#interface Tunnel0 — вход в конфигурирование интерфейса Tunnel0;

Router_A(config-if)#ip address 192.168.2.1 255.255.255.252 — назначение IP-адреса для интерфейса Tunnel0;

Router_A(config-if)#tunnel source 11.11.11.11 — указание на IP-адрес интерфейса, являющегося источником туннеля;

Router_A(config-if)#tunnel destination 33.33.33.33 — указание на IP-адрес окончания туннеля;

Router_A(config-if)#exit

Router_A(config)#ip route 192.168.3.0 255.255.255.0 Tunnel0 — добавление статического маршрута ко второй локальной сети LAN2;

Router_A(config)#ip route 0.0.0.0 0.0.0.0 11.11.11.12 — добавление статического маршрута в направлении внешнего интерфейса.

Команда **tunnel source 11.11.11.11** предполагает возможность в качестве адреса источника туннеля указать не только IP-адрес, но и указание на физический интерфейс:

Router_A(config-if)#tunnel source fa0/1

Более того, если в качестве симулятора использовать Cisco Packet Tracer, можно будет указать в качестве источника только физический интерфейс.

Команда **ip route 192.168.3.0 255.255.255.0 Tunnel0** в Cisco Packet Tracer также не может быть выполнена, вместо параметра **Tunnel0** необходимо использовать IP-адрес другого «конца» туннеля, в нашем случае 192.168.2.2.

Проведем аналогичную конфигурацию для маршрутизатора Router_B:

Router_B(config)#interface fastEthernet0/0

Router_B(config-if)#ip address 192.168.3.1 255.255.255.0

Router_B(config-if)#no shutdown

Router_B(config-if)#exit

Router_B(config)#interface fastEthernet0/1

Router_B(config-if)#ip address 33.33.33.33 255.0.0.0

Router_B(config-if)#no shutdown

Router_B(config-if)#exit

Router_B(config)#interface Tunnel0

Router_B(config-if)#ip address 192.168.2.2 255.255.255.252

Router_B(config-if)#tunnel source 33.33.33.33

Router_B(config-if)#tunnel destination 11.11.11.11

Router_B(config-if)#exit

Router_B(config)#ip route 192.168.1.0 255.255.255.0 Tunnel0

Router_B(config)#ip route 0.0.0.0 0.0.0.0 33.33.33.34

После успешного установления туннеля компьютеры из подсети 192.168.3.0/24 оказываются доступны из подсети 192.168.1.0/24, рисунок 6.3. Анализ содержимого пакета, передаваемого по туннелю, показан на рисунке 6.4.

Из рисунка 6.4 видно, что пакет с адресами источника и назначения 192.168.1.2 и 192.168.3.2 соответственно, вложен в пакет с адресами источника и назначения 11.11.11.11 и 33.33.33.33.

Интерфейс туннеля с номером 0 можно просмотреть на маршрутизаторе с использованием команды **show running-config**, рисунок 6.5.

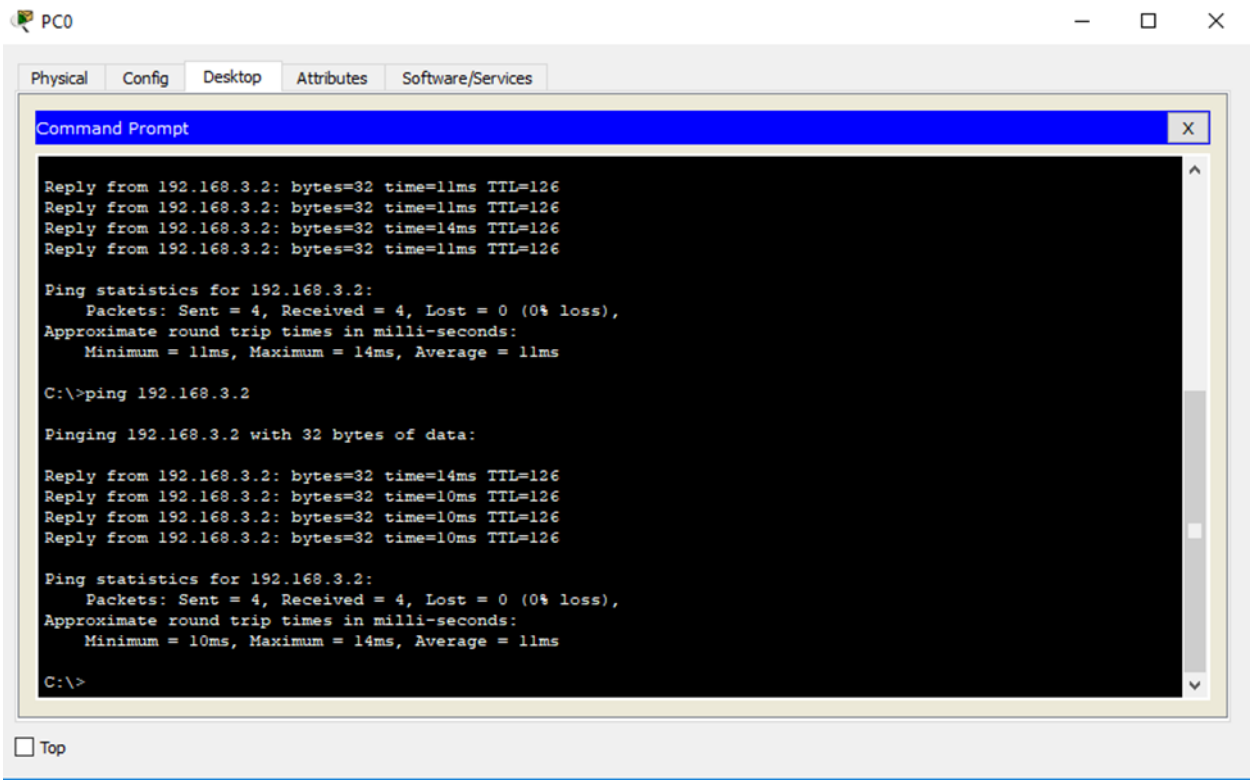


Рисунок 6.3 – Доступ ко второй локальной сети из первой с использованием туннеля

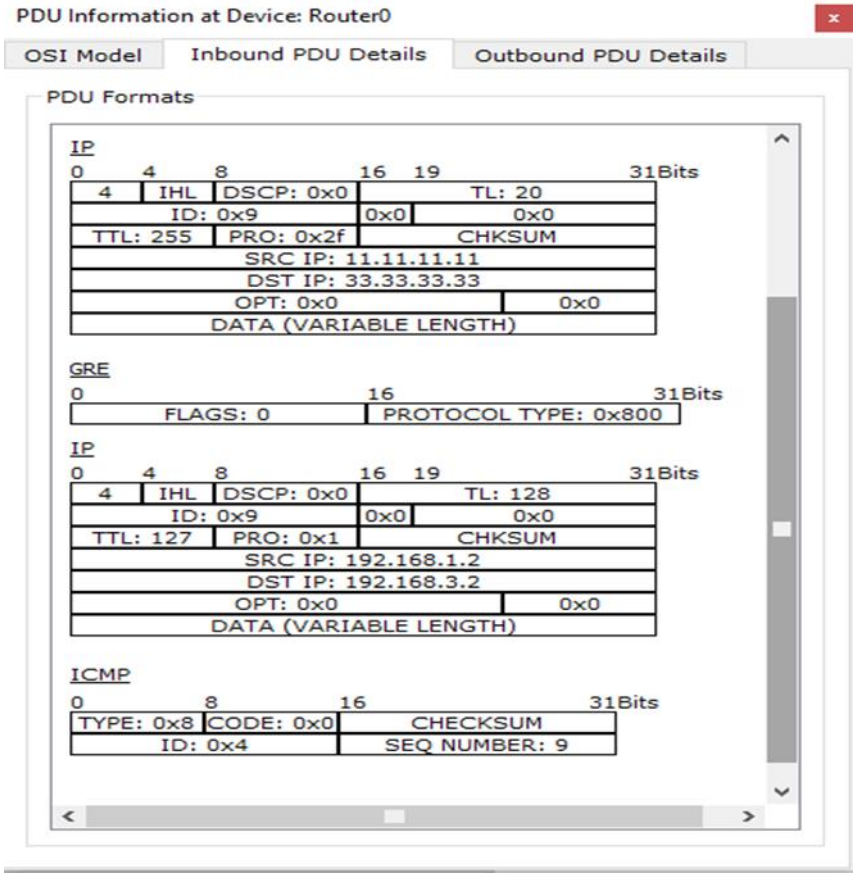


Рисунок 6.4 – Анализ содержимого пакета, передаваемого по туннелю

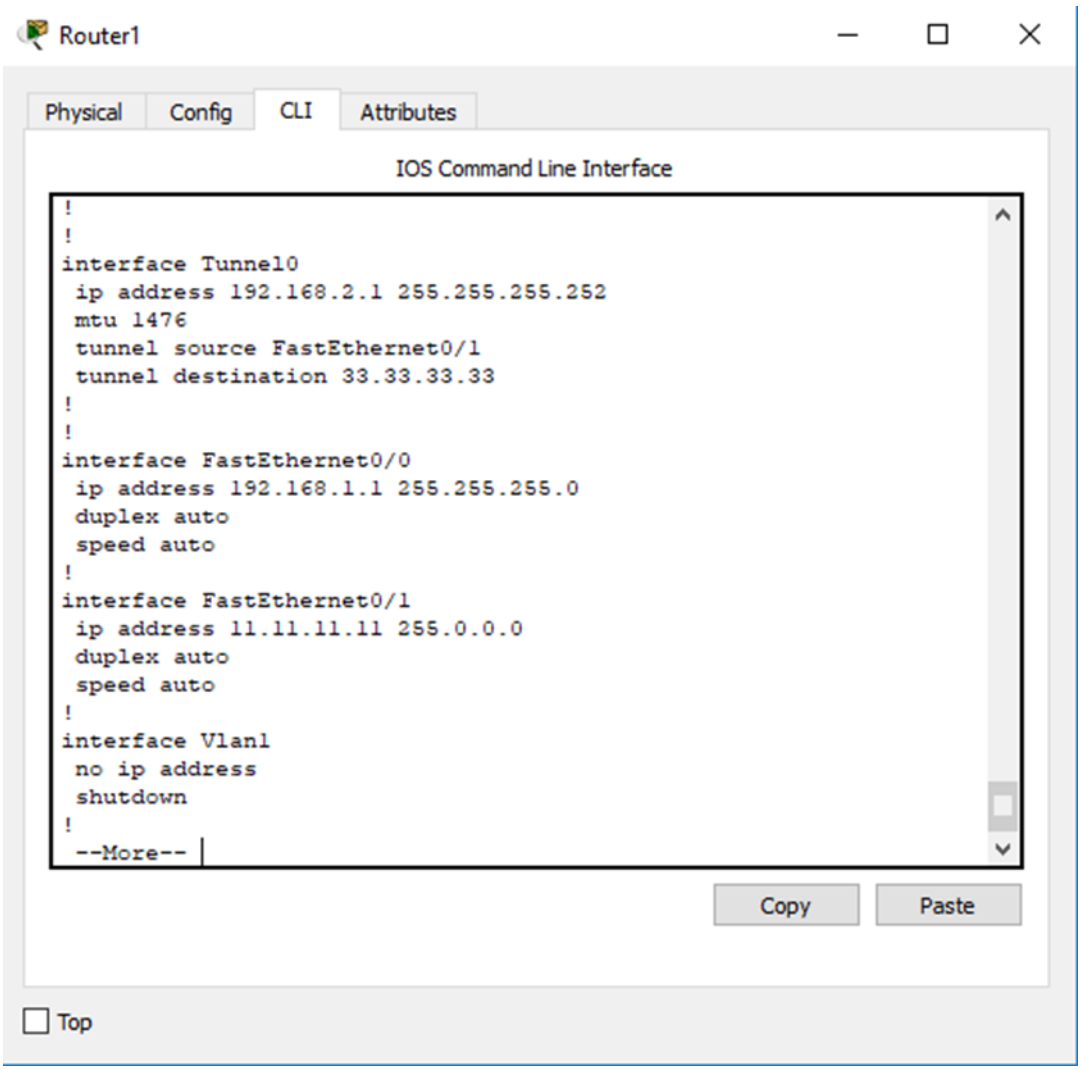


Рисунок 6.5 – Просмотр интерфейса tunnel0 на маршрутизаторе

Как указывалось выше, туннелирование само по себе не обеспечивает защиту передаваемых данных. Для этого используется семейство протоколов IP Security (IPSec), которые рассмотрим ниже.

6.2 Архитектура IPSec

IPSec – это комплект протоколов, касающихся вопросов шифрования, аутентификации и обеспечения защиты при транспортировке IP-пакетов. В его состав сейчас входят почти 20 предложений по стандартам и 18 RFC. Задача IPSec сводится к тому, чтобы выбрать конкретные алгоритмы и механизмы и настроить соответствующим образом устройства, участвующие в создании безопасного соединения.

При создании защищенного канала участникам данного процесса необходимо произвести следующие действия:

1. Аутентифицировать друг друга.
2. Сгенерировать и обменяться ключами.
3. Договориться о том, с помощью каких протоколов шифровать данные.
4. Начать передавать данные в зашифрованный туннель.

IPsec, как уже было указано ранее, состоит из нескольких протоколов, каждый из которых отвечает за конкретную стадию установления IPsec-туннеля. Обобщенная архитектура IPSec представлена на рисунке 6.6.

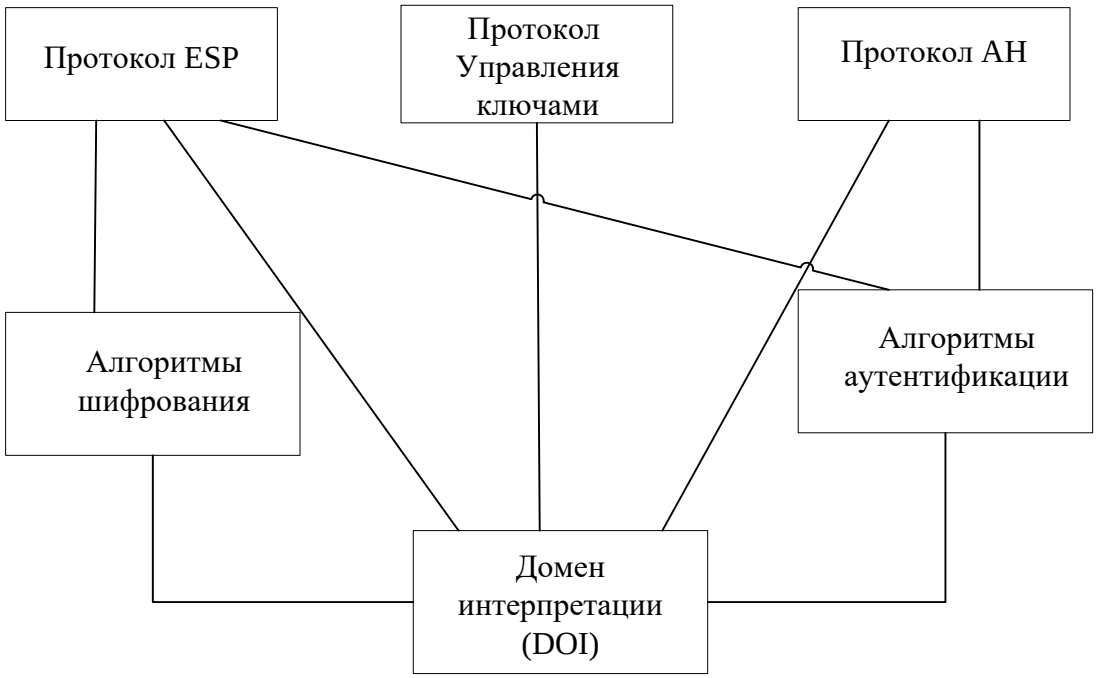


Рисунок 6.6 – Архитектура IPSec

Для управления ключами могут использоваться различные протоколы. Как известно, для шифрования и аутентификации необходимы ключи. Самым простым способом является ручная конфигурация ключей, однако такой способ плохо масштабируется. Поэтому на практике широко используется IKE – Internet Key Exchange — протокол обмена ключами, который позволяет участникам динамически аутентифицировать друг друга, согласовывать использование SA (об SA речь пойдет ниже), генерировать

ключи и обмениваться ими, используя алгоритм Диффи-Хеллмана (Приложение 1). IKE использует ISAKMP (Internet Security Association Key Management Protocol – протокол управления ключами ассоциации безопасности Интернет, разработан Национальным агентством безопасности США - NSA). Протокол ISAKMP сам по себе не регламентирует какой-либо конкретный алгоритм обмена ключами, он содержит в себе описание ряда сообщений, которые позволяют согласовать использование алгоритмов обмена ключами. IKE пришел в 1998 г. на смену более ранним протоколам — ISAKMP/Oakley.

Протокол ISAKMP , описанный в документе RFC 2408 [11], позволяет согласовывать алгоритмы и математические структуры (так называемые мультипликативные группы, определенные на конечном поле) для процедуры обмена ключами Диффи-Хеллмана, а также процессов аутентификации. Протокол Oakley, описанный в RFC 2412 [12], основан на алгоритме Диффи-Хеллмана и служит для организации непосредственного обмена ключами.

Для выполнения аутентификации сторон в IKE применяются два основных способа.

Первый способ основан на использовании разделяемого ключа. Перед инициализацией IPSec-устройств, образующих безопасные ассоциации, в их базы данных помещается предварительно распределенный разделяемый ключ. Цифровая подпись на основе односторонней хэш-функции, например, MD5, использующей в качестве аргумента этот предварительно распределенный ключ, доказывает аутентичность противоположной стороны.

Второй способ основан на использовании технологии цифровой подписи и цифровых сертификатов стандарта X.509. Каждая из сторон подписывает свой цифровой сертификат своим закрытым ключом и передает эти данные противоположной стороне. Если подписанный сертификат расшифровывается открытым ключом отправителя, то это удостоверяет тот факт, что отправитель, предоставивший данные, действительно обладает

ответной частью данного открытого ключа — соответствующим закрытым ключом.

Однако следует отметить, что для удостоверения аутентичности стороны нужно еще убедиться в аутентичности самого сертификата, и для этого сертификат должен быть подписан не только его владельцем, но и некоторой третьей стороной, выдавшей сертификат и вызывающей доверие. В архитектуре IPSec эта третья сторона именуется органом сертификации CA (Certification Authority). Этот орган призван засвидетельствовать подлинность обеих сторон и должен пользоваться полным доверием сторон, а его открытый ключ — известен всем узлам, использующим его сертификаты для удостоверения личностей друг друга.

Таким образом, IKE является комбинацией протоколов ISAKMP, Oakley и SKEME. Протокол ISAKMP описывает базовую технологию аутентификации, обмена ключами и согласования остальных параметров IPSec-туннеля при создании безопасного соединения, однако сами протоколы аутентификации сторон и обмена ключами в нем детально не определены. Поэтому при разработке протокола IKE общие правила и процедуры протокола ISAKMP дополнены процедурами аутентификации и обмена ключами, взятыми из протоколов Oakley и SKEME. Поскольку протокол IKE использует для управления ассоциациями алгоритмы и форматы протокола ISAKMP, названия этих протоколов иногда используют как синонимы.

Следующий протокол на рисунке 6.6 – AH (Authentication Header – заголовок аутентификации) отвечает за аутентификацию источника и проверку целостности данных.

И, наконец, протокол ESP (Encapsulating Security Payload – безопасная инкапсуляция полезной нагрузки) занимается непосредственно шифрованием данных, а также может обеспечивать аутентификацию источника и проверку целостности данных.

Здесь необходимо ввести в рассмотрение термин SA – Security Association. SA в общем смысле представляет собой набор параметров защищенного соединения (например, алгоритм шифрования, ключ шифрования, и т.д.), который может использоваться обеими сторонами соединения. У каждого соединения есть ассоциированный с ним SA. При этом SA носит односторонний характер, то есть для взаимодействия двух объектов между ними должно быть установлено, как минимум, две SA. Стандарты IPSec позволяют шлюзам использовать как одну ассоциацию SA для передачи трафика всех взаимодействующих через общедоступную сеть хостов, так и создавать для этой цели произвольное число ассоциаций SA, например, по одной на каждое соединение TCP.

Для идентификации каждой SA предназначен индекс параметров безопасности SPI (Security Parameters Index). Этот индекс включается в заголовки защищенных IPSec-пакетов, чтобы принимающая сторона смогла правильно их расшифровать и аутентифицировать, воспользовавшись указанной безопасной ассоциацией.

Рассмотрим, каким образом организуется защищенное соединение между участниками информационного обмена (например, между двумя пограничными маршрутизаторами локальных сетей, в терминах IPSec они называются шлюзами безопасности – Security Gateway).

1. Участникам надо договориться, какие алгоритмы/механизмы защиты они будут использовать для своего защищенного соединения, для чего используется протокол IKE. Этот процесс состоит из двух фаз:

1а) участники аутентифицируют друг друга и договариваются о параметрах установки вспомогательного соединения (тоже защищенного), предназначенного только для обмена информацией о желаемых/поддерживаемых алгоритмах шифрования и прочих деталях будущего IPSec-туннеля. Такой вспомогательный туннель называется ISAKMP-туннелем. Таким образом, первая фаза служит для создания

вспомогательного защищенного ISAKMP-туннеля, через который будут передаваться параметры для будущего IPSec-туннеля.

1б) уже доверяющие друг другу участники договариваются о том, как строить основной туннель для передачи данных. Они по очереди предлагают друг другу варианты и, если приходят к согласию, устанавливают основной туннель (IPSec-туннель).

2. Участники получили IPSec-туннель с параметрами, которые устраивают их обоих, и направляют туда потоки данных, подлежащие шифрованию.

3. Периодически, в соответствии с настроенным временным интервалом (Lifetime), обновляются ключи шифрования для основного туннеля. Для этого участники вновь связываются по ISAKMP-туннелю, проходят вторую фазу и устанавливают новые SA.

IPSec предлагает различные методы защиты трафика. В каждом узле, поддерживающем IPSec, используются базы данных (БД) двух типов:

- база данных безопасных ассоциаций SAD (Security Associations Database);
- база данных политики безопасности SPD (Security Policy Database).

При установлении SA две вступающие в обмен стороны принимают ряд соглашений, регламентирующих процесс передачи потока данных между ними. Соглашения представляются в виде набора параметров. Для SA такими параметрами являются, в частности, тип и режим работы протокола защиты (AH или ESP), методы шифрования, секретные ключи, значение текущего номера пакета в ассоциации и другая информация.

Объединение служебной информации в рамках SA предоставляет пользователю возможность сформировать разные классы защиты, предназначенные, например, для электронного общения с разными «собеседниками». Другими словами, применение структур SA открывает

путь к построению множества виртуальных частных сетей, различающихся своими параметрами.

Наборы текущих параметров, определяющих все активные ассоциации, хранятся на обоих оконечных узлах защищенного канала в виде SAD. Каждый узел IPSec поддерживает две базы SAD — одну для исходящих ассоциаций, другую — для входящих.

SPD задает соответствие между IP-пакетами и установленными для них правилами обработки. При обработке пакетов БД SPD используются совместно с БД SAD. SPD представляет собой упорядоченный набор правил, каждое из которых включает совокупность фильтров и допустимых политик безопасности. Фильтры служат для отбора пакетов, а политики безопасности задают требуемую обработку. Такая БД формируется и поддерживается на каждом узле, где реализуется IPSec.

Необходимо отметить, что первая фаза 1а) может проходить в одном из двух режимов – основном (main mode) и агрессивном (aggressive mode).

В основном режиме (main mode) первая фаза состоит из следующих этапов:

1. Обмен параметрами безопасности будущего ISAKMP-туннеля (алгоритмы шифрования, методы аутентификации, способ обмена секретными ключами, срок жизни SA);

2. Генерация каждым из участников общего секретного ключа;

3. Обмен общими секретными ключами с использованием алгоритма Диффи-Хеллмана;

4. Взаимная аутентификация участников.

В агрессивном режиме (aggressive mode) в первый пакет сразу помещается вся необходимая информация для установления ISAKMP-туннеля. Получатель посылает в ответ все, что необходимо для завершения обмена, после чего первому узлу необходимо лишь подтвердить соединение.

Агрессивный режим быстрей позволяет установить туннель, но при этом он менее безопасный, потому что стороны обмениваются информацией до того как безопасное соединение установлено.

6.3 Протокол AH

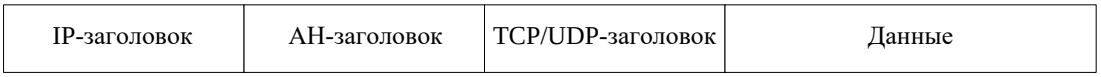
AH (Authentication Header) — протокол IPSec, предназначенный для аутентификации отправителя, контроля целостности данных и, опционально, для предотвращения атак в виде повторной посылки пакета (reply). По сути это обычный опциональный заголовок, располагающийся между основным заголовком IP-пакета и полем данных.

AH (как и рассматриваемый ниже ESP) может работать в одном из двух режимов – транспортном и туннельном. В первом случае (транспортный режим) механизмы безопасности применяются только для транспортного уровня и выше. Соответственно, заголовок сетевого уровня (IP) остается без защиты, более того, он остается таким же, как и был в исходном пакете, за исключением изменения некоторых полей. Например, меняется поле Next Header, указывающее на то, заголовок какого протокола следует за IP-заголовком.

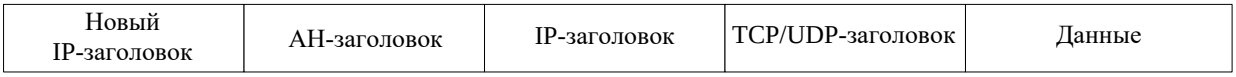
В туннельном режиме обеспечивается защита также и данных сетевого уровня. Это обеспечивается путем добавления нового IP-заголовка. После определения SA истинные адреса хостов отправления и назначения (и другие служебные поля) полностью защищаются от модификаций, а в новый заголовок выставляются адреса и другие данные для шлюзов отправления/получения.

На рисунке 6.7 показаны исходный пакет, пакеты после обработки протоколом AH в транспортном и туннельном режимах.

а) исходный пакет



б) пакет после обработки в транспортном режиме



в) пакет после обработки в туннельном режиме

Рисунок 6.7 – Форматы пакетов

Формат AH-заголовка показан на рисунке 6.8.

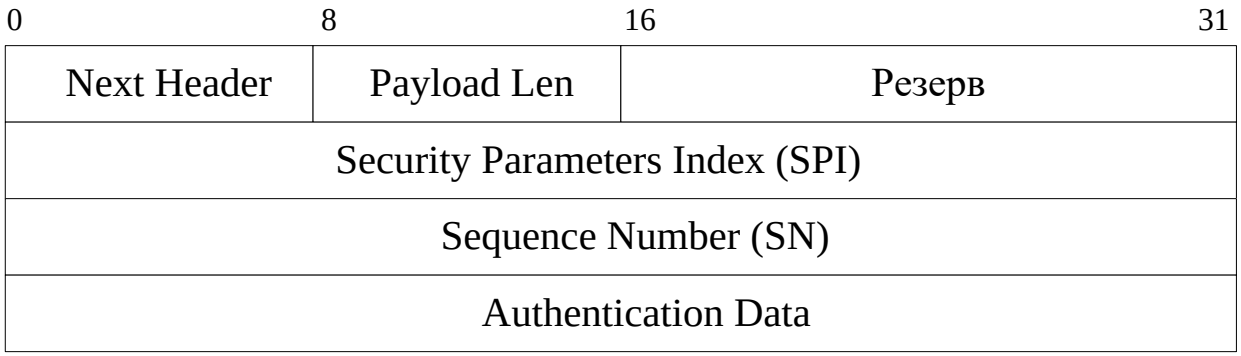


Рисунок 6.8 – Формат заголовка AH

Первые 8 бит заголовка (поле Next Header) содержат номер, соответствующий протоколу следующего уровня. Номер для каждого протокола назначает организация IANA (Internet Assigned Numbers Authority). Например, номер TCP - 6, ESP - 50, AH - 51 и т.д.

Поле Payload Len указывает длину заголовка AH в 32-битных словах. Необходимость этого поля связана с тем, что поле Authentication Data имеет переменную длину, соответственно, и сам AH-заголовок имеет переменную длину.

SPI – это 32-битный индикатор, который однозначно идентифицирует ту SA, в рамках которой передается данный пакет (об SPI уже шла речь выше).

Поле Sequence Number было введено в RFC 2402. Значение счетчика, содержащееся в этом поле, может использоваться для защиты от атак путем повторных посылок перехваченных пакетов. Если функция защиты от повторов активирована (а это указывается в SA), отправитель последовательно наращивает значение поля для каждого пакета, передаваемого в рамках данной SA.

Сама аутентификация обеспечивается передачей данных, содержащихся в поле Authentication Data. Аутентификационные данные представляют собой хэш-функцию, вычисленную на основе содержимого пакета с использованием алгоритмов MD5 или SHA-1. Симметричный секретный ключ шифрования устанавливается вручную или по протоколу IKE.

Аутентификация производится путем создания так называемой имитовставки (MAC), для чего используется хэш-функция и секретный ключ. Во всех реализациях AH обязательно должно поддерживаться использование хэш-функций с ключом HMAC-MD5-96 (используется по умолчанию) и HMAC-SHA-1-96, представляющих собой варианты хэш-функций MD5 и SHA-1, соответственно. Но могут использоваться и другие алгоритмы хэширования. Полученное значение, называемое в описании протокола ICV (Integrity Check Value - значение контроля целостности) помещается в поле Authentication Data. Это поле переменной длины, т.к. разные алгоритмы хеширования формируют разные по длине дайджесты.

При использовании AH в транспортном режиме, ICV рассчитывается для TCP/UDP-заголовка, данных и неизменяемых полей IP-заголовка. Изменяемые поля, такие например как поле TTL, при расчете значения хэш-функции принимаются равными 0. В туннельном режиме хэшируется весь исходный IP-пакет и неизменяемые поля нового заголовка.

6.4 Протокол ESP

ESP (Encapsulation Security Payload) — протокол IPSec, предназначенный для шифрования данных. ESP предоставляет три вида сервисов безопасности:

- обеспечение конфиденциальности (шифрование содержимого IP-пакетов, а также частичная защита от анализа трафика путем применения туннельного режима);

- обеспечение целостности IP-пакетов и аутентификации источника данных;

- обеспечение защиты от воспроизведения IP-пакетов.

Как видно, функциональность ESP шире, чем AH (добавляется шифрование). Кроме того, ESP не обязательно предоставляет все сервисы, но либо конфиденциальность, либо аутентификация должны быть задействованы.

Так же, как и AH, ESP может работать в транспортном и туннельном режимах. Исходный пакет, пакеты после обработки протоколом ESP в транспортном и туннельном режимах показаны на рисунке 6.9. На рисунке серым фоном отмечены поля пакета, подлежащие шифрованию.

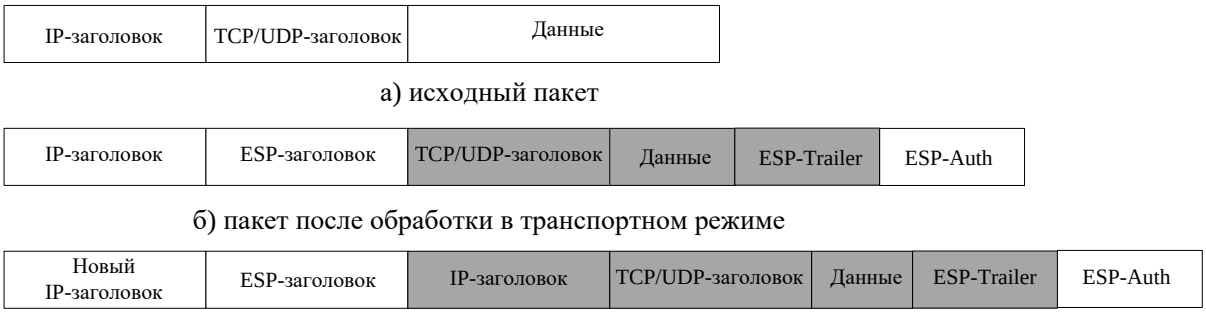


Рисунок 6.9 – Форматы пакетов

Так как ESP поддерживает и аутентификацию, и шифрование, для аутентификации может использоваться имитовставка с использованием хэш-

функций с ключом HMAC-MD5-96 (по умолчанию) и HMAC-SHA-1-96 (как и у AH), но могут использоваться и другие алгоритмы. Для шифрования могут использоваться алгоритмы DES, 3DES, AES, и т.д.

Формат заголовка ESP выглядит в соответствии с рисунком 6.10. Это не столько заголовок, сколько обертка (инкапсулирующая оболочка, как видно из рисунка 6.9) для зашифрованного содержимого. Например, ссылку на следующий заголовок нельзя выносить в начало, в незашифрованную часть, так как она лишится конфиденциальности.

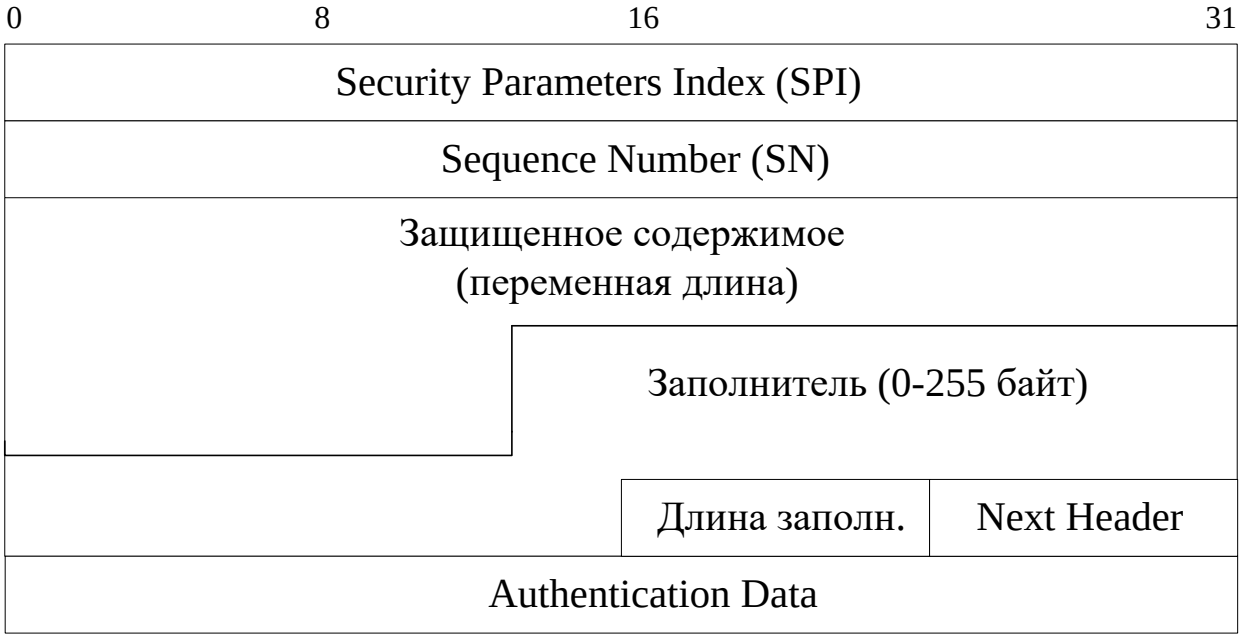


Рисунок 6.10 – Формат заголовка ESP

Заголовок ESP начинается с двух 32-разрядных значений - SPI и SN. Роль их такая же, как в протоколе AH - SPI идентифицирует SA, использующийся для создания данного туннеля; SN позволяет защититься от повторов пакетов. SN и SPI не шифруются.

Следующим идет поле, содержащее зашифрованные данные. После них следует поле заполнителя, который нужен для того, чтобы выровнять длину шифруемых полей до значения, кратного размеру блока алгоритма шифрования.

После заполнителя идут поля, содержащие значение длины заполнителя и указание на номер, соответствующий протоколу следующего уровня. Четыре перечисленных поля (данные, заполнитель, длина, следующий протокол) защищаются шифрованием.

Если ESP используется и для аутентификации данных, то завершает пакет поле переменной длины, содержащее ICV.

Применение протокола ESP к исходящим пакетам можно представлять себе следующим образом. Из пакета извлекается его часть – остаток, то есть та часть, которая подлежит шифрованию (закрашенные поля на рисунке 6.9). Далее следуют этапы:

- остаток пакета копируется в буфер;

- к остатку приписываются дополняющие байты, их число и номер (тип) первого заголовка остатка, с тем, чтобы номер был прижат к границе 32-битного слова, а размер буфера удовлетворял требованиям алгоритма шифрования;

- текущее содержимое буфера шифруется;

- в начало буфера приписываются поля "Индекс параметров безопасности (SPI)" и "Порядковый номер (SN)" с соответствующими значениями;

- пополненное содержимое буфера аутентифицируется, в его конец помещается поле "Аутентификационные данные";

- в новый пакет переписываются начальные заголовки старого пакета и конечное содержимое буфера.

Если в ESP включены и шифрование, и аутентификация, то аутентифицируется зашифрованный пакет. Для входящих пакетов действия выполняются в обратном порядке, т. е. сначала производится аутентификация. Это позволяет не тратить ресурсы на расшифровку поддельных пакетов, что в какой-то степени защищает от атак на доступность.

Из сравнения протоколов AH и ESP можно сделать следующие выводы. Если необходимо знать, что данные из идентифицированного источника передаются без нарушения целостности, а их конфиденциальность обеспечивать не требуется, можно использовать протокол AH, который защищает протоколы высших уровней и поля заголовка IP, не изменяемые в пути. Защита означает, что соответствующие значения нельзя изменить, потому что это будет обнаружено второй стороной IPSec, и любая модифицированная дейтаграмма IP будет отвергнута. Протокол AH не обеспечивает защиту от прослушивания канала и просмотра нарушителем заголовка и данных. Но поскольку заголовок и данные незаметно изменить нельзя, измененные пакеты отвергаются.

Если необходимо сохранить данные в тайне (обеспечить конфиденциальность), необходимо использовать ESP. Данный протокол предполагает шифрование протоколов высших уровней в транспортном режиме и всей исходной дейтаграммы IP в туннельном режиме, так что извлечь информацию о пакетах путем прослушивания канала передачи невозможно. Протокол ESP может также обеспечить для пакетов сервис аутентификации. Однако при использовании ESP в транспортном режиме внешний оригинальный заголовок IP не защищается, а в туннельном режиме не защищается новый заголовок IP.

Таким образом, при необходимости обеспечения конфиденциальности и целостности передаваемых данных с одновременной аутентификацией их источника необходимо использовать протокол ESP в туннельном режиме. Более того, при создании ISAKMP-туннеля необходимо использовать основной режим (main mode).

6.5 Конфигурирование VPN IPSec между маршрутизаторами Cisco

При использовании оборудования Cisco для создания VPN можно либо использовать рассмотренное выше туннелирование, и отправлять по

туннелю зашифрованный трафик, либо не использовать туннельные интерфейсы.

Рассмотрим конфигурирование VPN с туннелированием. Часто такой режим называют IPSec-over-GRE. Будем использовать тот же пример, рисунок 6.11.

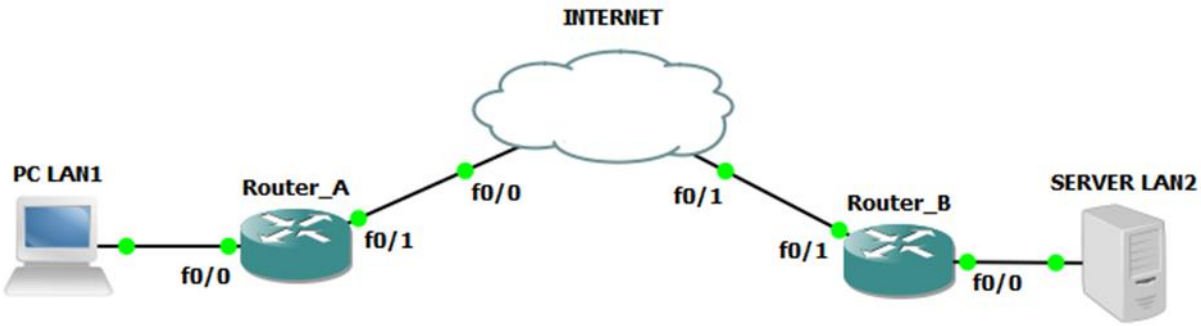


Рисунок 6.11 – Пример объединения двух частных сетей

Адресация в сети следующая:

Router_A:

f0/0 – 192.168.1.1/24

f0/1 – 11.11.11.11/8

Router_B:

f0/0 – 192.168.3.1/24

f0/1 – 33.33.33.33/8

Адрес туннеля – 192.168.2.0/30.

На первом этапе необходимо создать политику с номером 1 которая определит алгоритм и используемые протоколы при обмене ключами (IKE Фаза 1). Номер политики определяет ее приоритет, так как может быть создано несколько политик (чем меньше число, тем выше приоритет). Часто задаются несколько таких политик с различными комбинациями шифрования, хеша и номера группы Дифи-Хеллмана (DH). При создании isakmp sa, та сторона, которая инициирует соединение, отправляет все локально настроенные политики. Принимающая сторона просматривает по очереди, в порядке приоритетности, свои локально настроенные политики.

Первая же политика, для которой найдено совпадение, будет использоваться. Команда создания политики:

RouterA(config)#crypto isakmp policy 1

В соответствии с логикой Cisco IOS, если мы создаем что-либо, мы сразу попадаем в конфигурирование того, что мы создали. Поэтому после создания политики 1 мы попадаем в режим ее конфигурирования. Команды конфигурирования:

RouterA(config-isakmp)#encryption <алгоритм шифрования>

RouterA(config-isakmp)#hash <алгоритм хэширования>

RouterA(config-isakmp)#authentication <метод аутентификации>

RouterA(config-isakmp)#lifetime <время жизни ISAMKP-туннеля>

В качестве алгоритмов шифрования могут быть указаны **des, 3des, aes.** В качестве алгоритма хэширования может быть использован **sha** или **md5**. В качестве метода аутентификации может быть указан метод с открытым ключом **pre-share**, либо более сложные методы, например, аутентификация с использованием цифровой подписи RSA Digital Signatures **rsa-sig.**

При использовании **pre-share** может также быть задан номер группы Диффи-Хеллмана (DH):

RouterA (config-isakmp)# group 2

Группа означает конкретный алгоритм DH [13]:

Алгоритм DH-1 - ключ 768 бит;

Алгоритм DH-2 - ключ 1024 бит;

Алгоритм DH-5 - ключ 1536 бит.

Также в случае использования открытого ключа необходимо указать сам ключ (он должен быть одинаковым на обоих маршрутизаторах):

RouterA(config)#crypto isakmp key <№> <ключ> address <адрес соседа>

Далее указываются возможные параметры создаваемого SA (политика Фазы 2). Так как это набор протоколов, предлагаемых встречной стороне, он называется **transform-set**. Дальнейшее конфигурирование удачно иллюстрируется рисунком 6.12, заимствованным из [14].

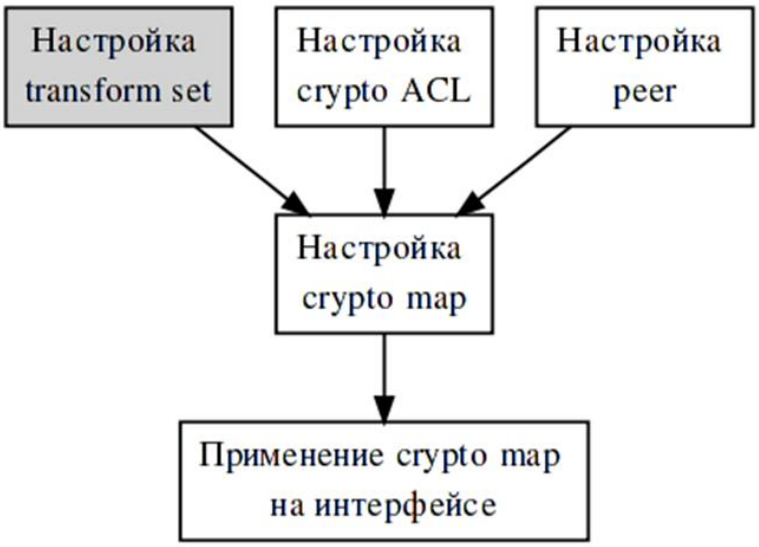


Рисунок 6.12 – Принцип конфигурирования transform-set

Из рисунка 6.12 видно, что сначала необходимо создать transform-set с указанием используемых протоколов (AH или ESP), способов аутентификации и шифрования. С использованием ACL фильтруется трафик, подлежащий шифрованию, с использованием peer указывается «сосед», с которым устанавливается туннель. Криптокарта (crypto map) представляет собой указание деталей шифрования – адрес «соседа» по туннелю, используемый ACL, используемый transform-set. Заключительным этапом является привязка созданной криптокарты на конкретном интерфейсе.

RouterA(config)# crypto ipsec transform-set <имя> <аутентификация> <шифрование>

ESP и AH могут быть использованы одновременно, а могут и поодиночке. В таблице 6.1 представлены каждый из возможных вариантов.

Таблица 6.1 – Варианты использования протоколов для AH и ESP

Тип преобразования	Синтаксис	Описание
AH Transform (один из списка)	ah-md5-hmac	Протокол AH с алгоритмом аутентификации MD5
	ah-sha-hmac	Протокол AH с алгоритмом аутентификации SHA
	ah-gost-28147-mac	Протокол AH с алгоритмом ГОСТ 28147-89 (в режиме

		выработки имитовставки)
	ah-gost3411-hmac	Протокол AH с алгоритмом ГОСТ Р 34.11-94
ESP Encryption Transform (один из списка)	esp-null	Протокол ESP с алгоритмом Null.
	esp-des	Протокол ESP с 56-битным алгоритмом DES
	esp-3des	Протокол ESP с 168-битным алгоритмом 3DES
	esp-aes-128	Протокол ESP с 128-битным алгоритмом AES
	esp-aes-192	Протокол ESP с 192-битным алгоритмом AES
	esp-aes-256	Протокол ESP с 256-битным алгоритмом AES
	esp-gost28147	Протокол ESP с алгоритмом ГОСТ 28147-89 (в режиме простой замены с зацеплением)
	esp-gost28147-4m-imit	Протокол ESP с алгоритмом ГОСТ 28147-89 (в комбинированном режиме: гаммирование и вычисление имитовставки. в соответствии со спецификацией ESP_GOST-4M-IMIT)
ESP Authentication Transform (один из списка)	esp-md5-hmac	Протокол ESP с алгоритмом аутентификации MD5
	esp-sha-hmac	Протокол ESP с алгоритмом аутентификации SHA
	esp-gost28147-mac	Протокол ESP с алгоритмом ГОСТ 28147-89 (в режиме выработки имитовставки)
	esp-gost3411-hmac	Протокол ESP с алгоритмом ГОСТ Р 34.11-94

Таким образом, если мы предполагаем использование только ESP с алгоритмом шифрования 3DES и аутентификацией с хэш-функцией MD5, команда принимает вид:

RouterA(config)#crypto ipsec transform-set LAN1 esp-3des esp-md5-hmac

Далее с использованием команд **mode tunnel** и **mode transport** можно включить либо туннельный, либо транспортный режимы.

Далее необходимо сформировать так называемую криптокарту (crypto map), в которой указываются детали шифрования:

RouterA(config)#crypto map <имя> <№> ipsec-isakmp

RouterA(config-crypto-map)#set peer <адрес соседа>

RouterA(config-crypto-map)#set transform-set <имя transform-set >

RouterA(config-crypto-map)#match address <список доступа>

В первой команде № - это номер набора шифрования для создаваемого туннеля. Дело в том, что в маршрутизаторе может существовать только одна криптокарта. Однако в самой карте можно создавать наборы шифрования для нескольких туннелей. Например, мы создаем второй туннель. Тогда в самой карте поменяем порядковый номер:

RouterA (config)# crypto map MAP <другой №> ipsec-isakmp

После этой команды можно указывать параметры для туннеля с номером **другой №**.

Также необходимо настроить список доступа (crypto ACL на рисунке 6.12). В данном случае шифровать необходимо те пакеты, которые передаются в туннель, то есть имеют GRE-заголовок. Поэтому создадим расширенный список доступа:

RouterA(config)#access-list 100 permit gre host 11.11.11.11 host 33.33.33.33

Теперь необходимо перейти в режим конфигурирования внешнего интерфейса (у нас это fa0/1) и привязать к нему созданную криптокарту:

RouterA(config)#interface fa 0/1

RouterA(config-if)#crypto map <имя кпиптокарты>

Аналогичные настройки необходимо произвести на втором шлюзе безопасности (у нас это RouterB):

RouterB(config)#crypto isakmp policy 1

RouterB(config-isakmp)#encryption 3des

RouterB(config-isakmp)#hash md5

RouterB(config-isakmp)#authentication pre-share

RouterB (config-isakmp)#group 2

RouterB(config)#exit

RouterB(config)#crypto isakmp key 0 PASS address 11.11.11.11

RouterB(config)#crypto ipsec transform-set LAN2 esp-3des esp-md5-hmac

RouterB(cfg-crypto-trans)#mode tunnel

RouterB(cfg-crypto-trans)#exit

RouterB(config)#crypto map MAP2 10 ipsec-isakmp

RouterB(config-crypto-map)#set peer 11.11.11.11

RouterB(config-crypto-map)#set transform-set LAN2

RouterB(config-crypto-map)#match address 100

RouterB(config-crypto-map)#exit

RouterB(config)#access-list 100 permit gre host 33.33.33.33 host 11.11.11.11

RouterB(config)#interface fa 0/1

RouterB(config-if)#crypto map MAP2

Настроенные параметры на Router_A можно просмотреть с использованием команды **show running-config,** рисунок 6.13.

```
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 lifetime 10000
crypto isakmp key PASS address 33.33.33.33
!
!
crypto ipsec transform-set FILIAL esp-3des esp-md5-hmac
!
crypto map MAP 10 ipsec-isakmp
 set peer 33.33.33.33
 set transform-set FILIAL
 match address 100
!
!
!
!
interface Tunnel0
 ip address 192.168.2.1 255.255.255.252
 tunnel source 11.11.11.11
 tunnel destination 33.33.33.33
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 11.11.11.11 255.0.0.0
 duplex auto
 speed auto
 crypto map MAP
```

Рисунок 6.13 – Просмотр конфигурации

Для просмотра параметров шифрования можно использовать ряд команд:

show crypto isakmp policy – просмотр параметров туннеля в первой фазе;

show crypto map – просмотр созданных карт шифрования;

show crypto isakmp sa – просмотр созданных SA (фаза 1);

show crypto ipsec sa – просмотр созданных SA (фаза 2).

Выполнение этих команд, выполненных с использованием GNS3, иллюстрируется рисунками 6.14 – 6.17.

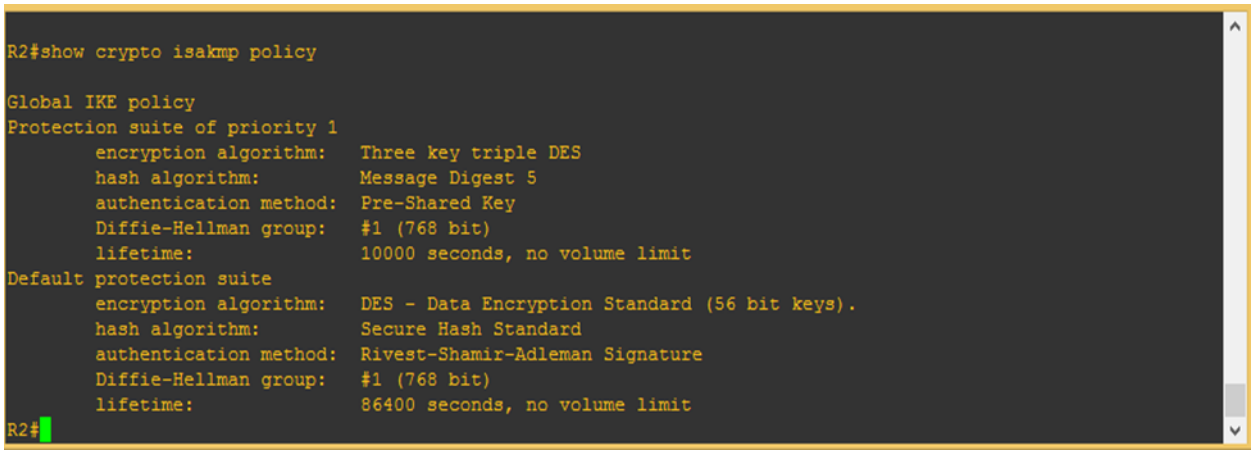


Рисунок 6.14 – Результат выполнения команды **show crypto isakmp policy**

Из рисунка видно, что на маршрутизаторе имеются две политики – одна созданная нами (1), и вторая – по умолчанию.

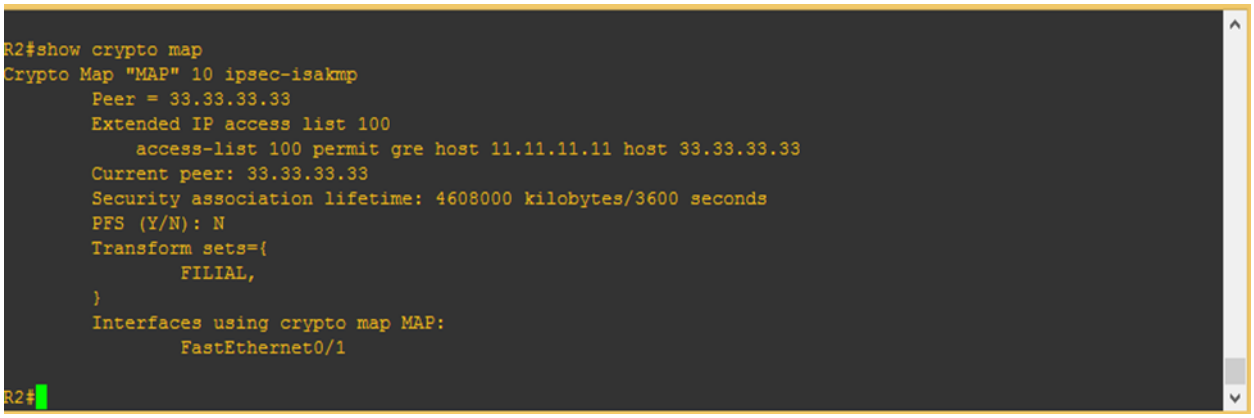


Рисунок 6.15 – Результат выполнения команды **show crypto map**

Рисунок 6.16 – Результат выполнения команды **show crypto isakmp sa**

Если в графе state указано QM_IDLE, значит первая фаза прошла успешно, в противном случае отобразится сообщение MM_NO_STATE. Также SA, созданные на первой фазе, можно просмотреть более подробно с использованием команды **show crypto isakmp sa detail**.

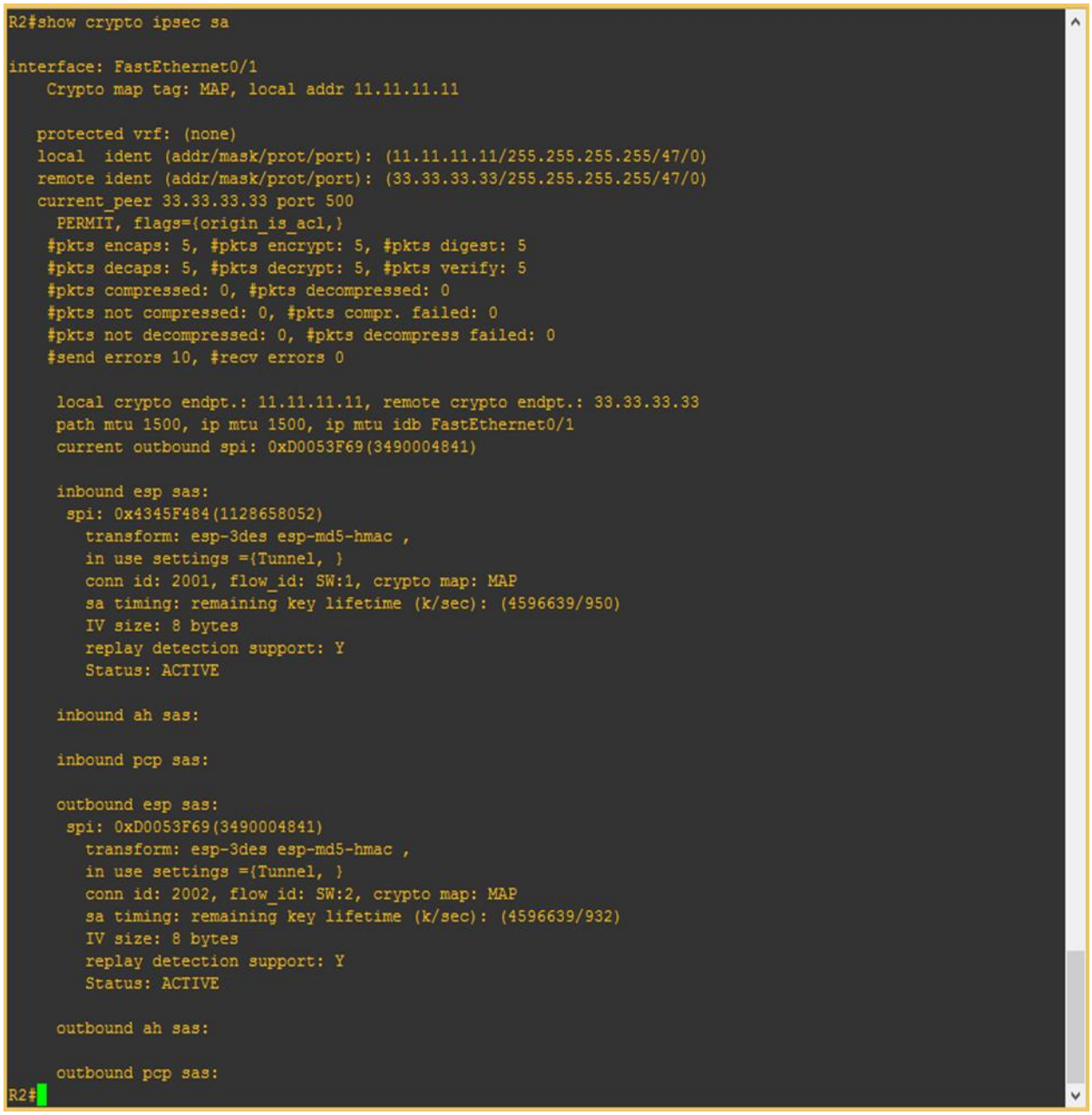


Рисунок 6.17 – Результат выполнения команды **show crypto ipsec sa**

Количество зашифрованных и расшифрованных пакетов можно просмотреть с использованием команды **show crypto engine connections active**, рисунок 6.18.

```
R2#show crypto engine connections active

  ID Interface           IP-Address      State  Algorithm           Encrypt  Decrypt
   1 FastEthernet0/1     11.11.11.11     set    HMAC_MD5+3DES_56_C        0        0
2001 FastEthernet0/1     11.11.11.11     set    3DES+MD5                  0        5
2002 FastEthernet0/1     11.11.11.11     set    3DES+MD5                  5        0

R2#
```

Рисунок 6.18 – Результат выполнения команды **show crypto engine connections active**

Второй способ позволяет настроить VPN без туннеля GRE, туннелирование производится самим IPSec. Для этого в списке доступа необходимо указать не протокол GRE, а в явном виде указать сети источника и получателя трафика:

RouterA(config)#ip access-list extended <ИМЯ>

permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255

Считаем, что сеть имеет вид, показанный на рисунке 6.19. На маршрутизаторах R0 и R2 должны быть заданы маршруты по умолчанию к R1.

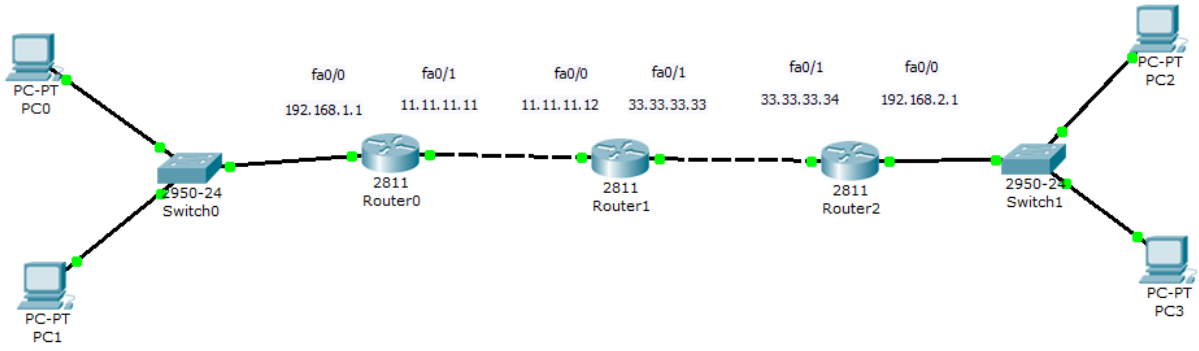


Рисунок 6.19 – Пример сети

Команды конфигурирования для Router_A в этом случае имеют вид:

RouterA(config)#crypto isakmp policy 1

RouterA(config-isakmp)#encryption 3des

RouterA(config-isakmp)#hash md5

RouterA(config-isakmp)#authentication pre-share

RouterA(config-isakmp)#group 2

RouterA(config)#exit

RouterA(config)#crypto isakmp key PASS address 33.33.33.34

RouterA(config)#crypto ipsec transform-set LAN esp-3des esp-md5-hmac

RouterA(config)#ip access-list extended FOR-VPN

RouterA(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255

RouterA(config-ext-nacl)#exit

RouterA(config)#crypto map MAP 10 ipsec-isakmp

RouterA(config-crypto-map)#set peer 33.33.33.34

RouterA(config-crypto-map)#set transform-set LAN

RouterA(config-crypto-map)#match address FOR-VPN

RouterA(config-crypto-map)#exit

RouterB(config)#interface fa 0/1

RouterB(config-if)#crypto map MAP

6.6 Практическое задание

6.6.1 Построить сеть, рисунок 6.19, с адресацией в «левой» сети 192.168.X.0, а в «правой» - 192.168.X+1.0. Прописать статические маршруты по умолчанию.

6.6.2 Сконфигурировать VPN между локальными сетями.

6.6.3 Проверить работоспособность VPN-канала.

Заключение

В учебном пособии рассмотрены основные методы и средства, необходимые для предотвращения несанкционированного доступа к передаваемым данным, а также их конфигурирование на примере оборудования Cisco Systems. Учебный материал, изложенный в пособии, является базой для обучения студентов принципам построения защищенных сетей связи. Кроме изучения дисциплин учебного плана по направлению 11.03.02 «Инфокоммуникационные технологии и системы связи», данное пособие может быть использовано при подготовке выпускной квалификационной работы бакалавра.

Список использованных источников

1. Манин А.А., Сосновский И.А. Системы коммутации. Принципы и технологии пакетной коммутации. Учебное пособие. Издание 2-е, переработанное и дополненное. – Ростов-на-Дону, 2017.

2. Белов Е.Б., Лось В.П., Мещеряков Р.В., Шелупанов А.А. Основы информационной безопасности. Учебное пособие для ВУЗов, 2006.

3. https://tools.ietf.org/html/rfc854

4. https://tools.ietf.org/html/rfc4251

5. https://www.wireshark.org/

6. https://tacacsgui.com/

7. Руководящий документ. Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. Утверждено решением председателя Государственной технической комиссии при Президенте Российской Федерации от 25 июля 1997 г.

8. https://www.cisco.com/c/ru_ru/products/collateral/switches/catalyst-2960-x-series-switches/data_sheet_c78-728232.pdf

9. https://tools.ietf.org/html/rfc2784

10. http://yztm.ru/2018/03/18/vpncisco1/

11. https://tools.ietf.org/html/rfc2408

12. https://tools.ietf.org/html/rfc2412

13. https://techprofi.com/network/nastraivaem-vpn-ipsec-cisco/

14. http://xgu.ru/wiki/IPsec_%D0%B2_Cisco