МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ
Северо-Кавказский филиал ордена Трудового Красного Знамени
федерального государственного бюджетного образовательного учреждения
высшего образования
«Московский технический университет связи и информатики»

Методические указания
к лабораторным работам
по дисциплине
«Методы и средства защиты компьютерной информации»

(направление подготовки 09.03.01 «Информатика и вычислительная техника»)

Ростов-на-Дону
2022

Методические указания

к лабораторным работам

по дисциплине

«Методы и средства защиты компьютерной информации»

Составители:

Манин А.А., доцент кафедры «Инфокоммуникационные технологии и системы связи», к.т.н., доцент

Рассмотрено и одобрено

на заседании кафедры ИВТ

Протокол № 5 от 19.12.2022 г.

Лабораторная работа 1. Исследование свойств межсетевого экрана с пакетной фильтрацией

1.1 Цель работы: Получение навыков конфигурирования межсетевого экрана с пакетной фильтрацией и исследование его свойств.

1.2 Перечень оборудования:

- Маршрутизаторы Cisco;
- ПК с установленным ПО Cisco Packet Tracer;
- Локальная сеть.

1.3 Задание:

- 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.
- Всем компьютерам подсети 192.169.X.0 (где X – номер студента в списке группы) предоставить полный доступ ко всем серверам.
- Всем компьютерам подсети 192.169.X+1.0 предоставить доступ только к FTP-серверу по протоколу FTP.
- Всем компьютерам подсети 192.169.X+2.0 предоставить доступ только к WEB-серверу.
- Компьютерам оставшейся подсети запретить доступ к внешним ресурсам.

1.4 Указания к проведению работы.

Как известно [7], разделяют межсетевые экраны (FireWall) с пакетной фильтрацией и с сохранением состояний (экраны прикладного уровня). Рассмотрим сначала использование пакетной фильтрации.

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. 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 1.1).

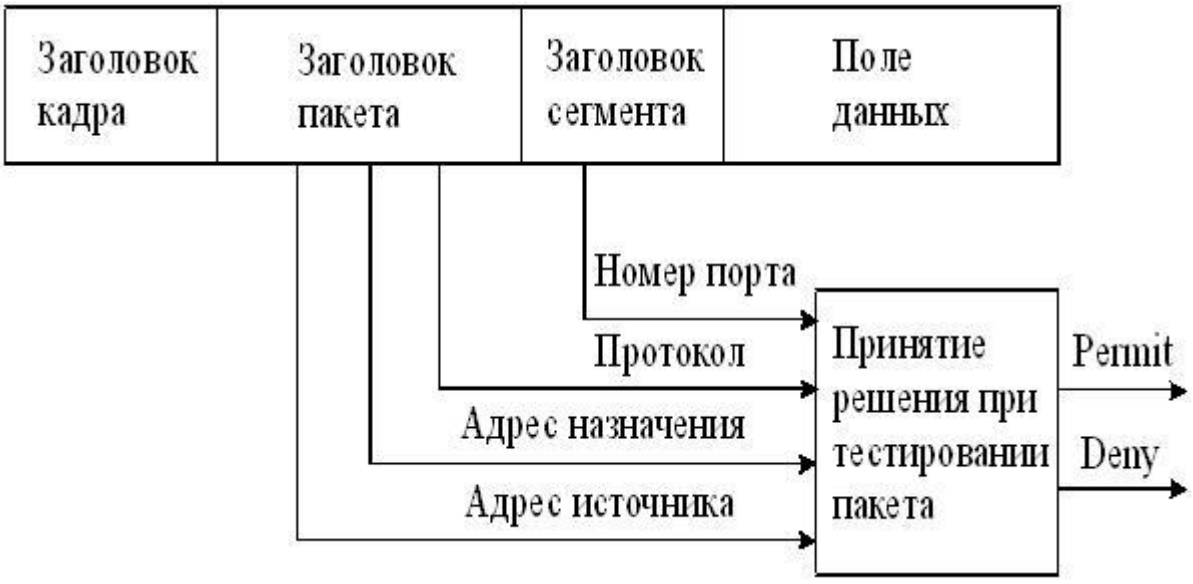

Рисунок 1.1 – Анализ заголовков пакета

Как видно из рисунка 1.1, 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 IOS Cisco 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 1.1).

Таблица 1.1 – 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 1.2.

Таблица 1.2 – 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 1.3.

Таблица 1.3 – 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 1.2. На рисунке укажем узлы, находящиеся в подсетях 192.168.0.0/24 и 192.168.1.0/24.

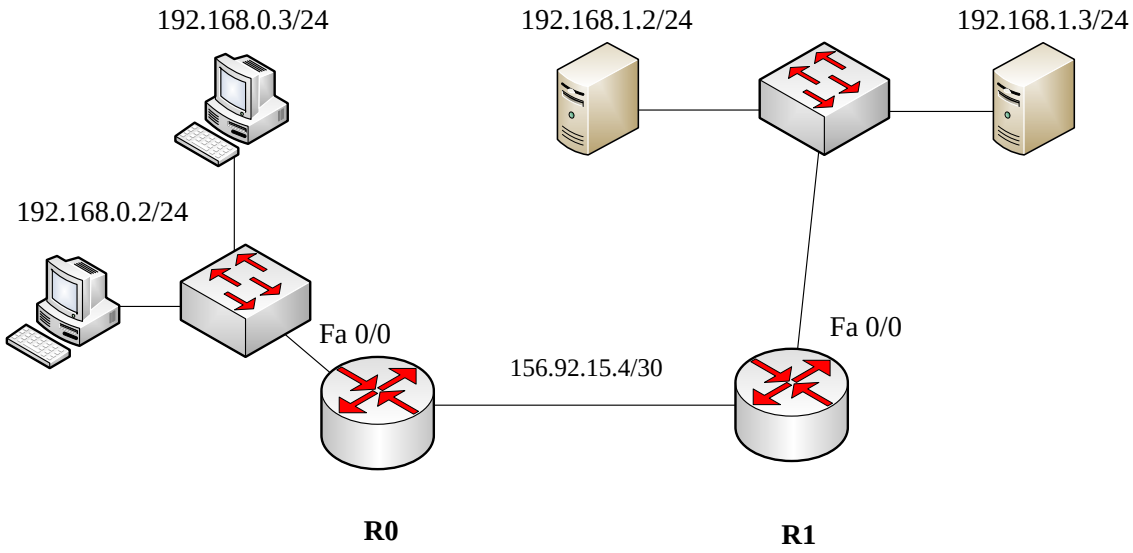

Рисунок 1.2 – Схема сети

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 3.2), 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 1.3 показаны настроенные списки доступа для рассмотренного здесь примера.

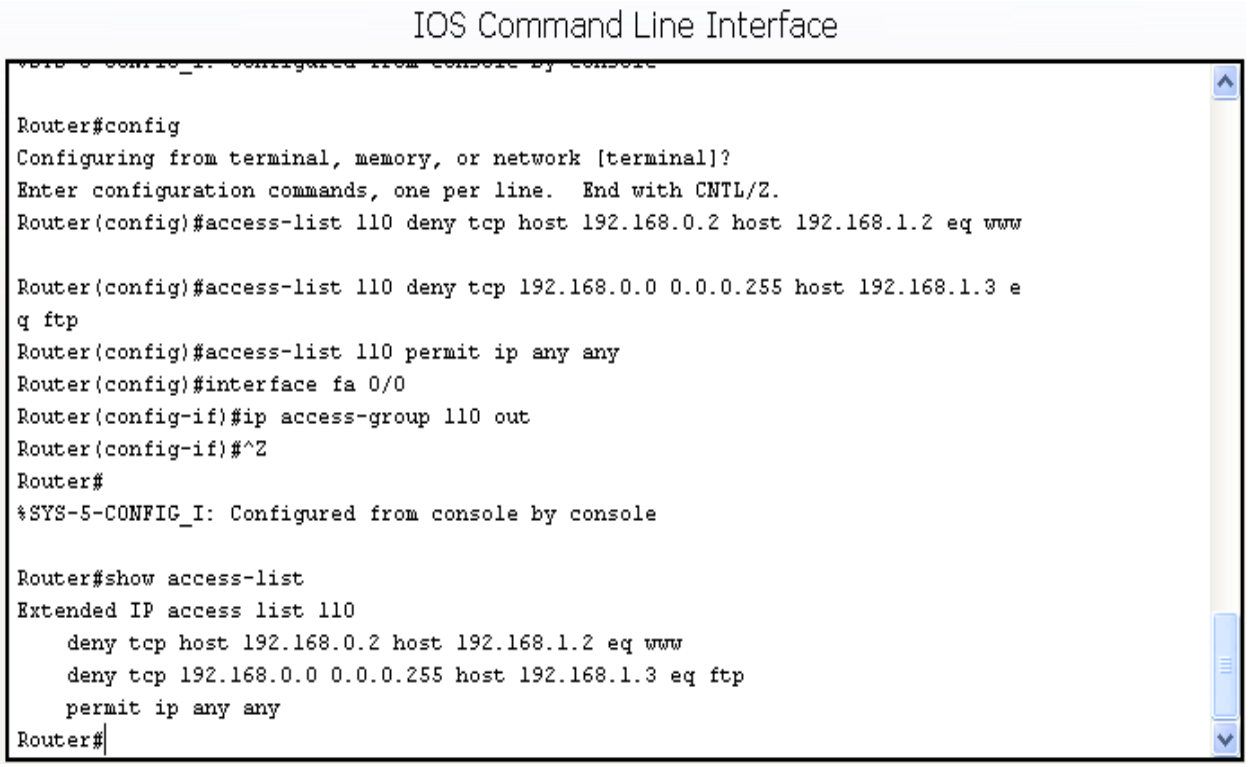

Рисунок 1.3 – Конфигурирование списка доступа и его просмотр

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.

1.5 Отчет по работе:

- Демонстрация функционирования МСЭ.

Лабораторная работа 2. Исследование свойств межсетевого экрана с сохранением состояний

2.1 Цель работы: Получение навыков конфигурирования межсетевого экрана с сохранением состояний и исслежование его свойств.

2.2 Перечень оборудования:

- Маршрутизаторы Cisco;
- ПК с установленным ПО Cisco Packet Tracer;
- Локальная сеть.

2.3 Задание:

- 3.4.1 Используя Cisco Packet Tracer, построить сеть, показанную на рисунке 3.4. Произвести конфигурирование сетевых устройств для обеспечения доступа всех серверов из внутренней сети.
- 3.4.2 Убедиться в возможности доступа во внутреннюю сеть извне с использованием произвольного протокола.
- 3.4.3 Настроить инспектирование TCP-трафика и сконфигурировать список доступа. Убедиться в доступности серверов из внутренней сети и недоступности ресурсов внутренней сети извне.

2.4 Указания по проведению работы

Такие межсетевые экраны еще называют экранами с сохранением сессий (statefull firewall). 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 2.1. 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 2.2).

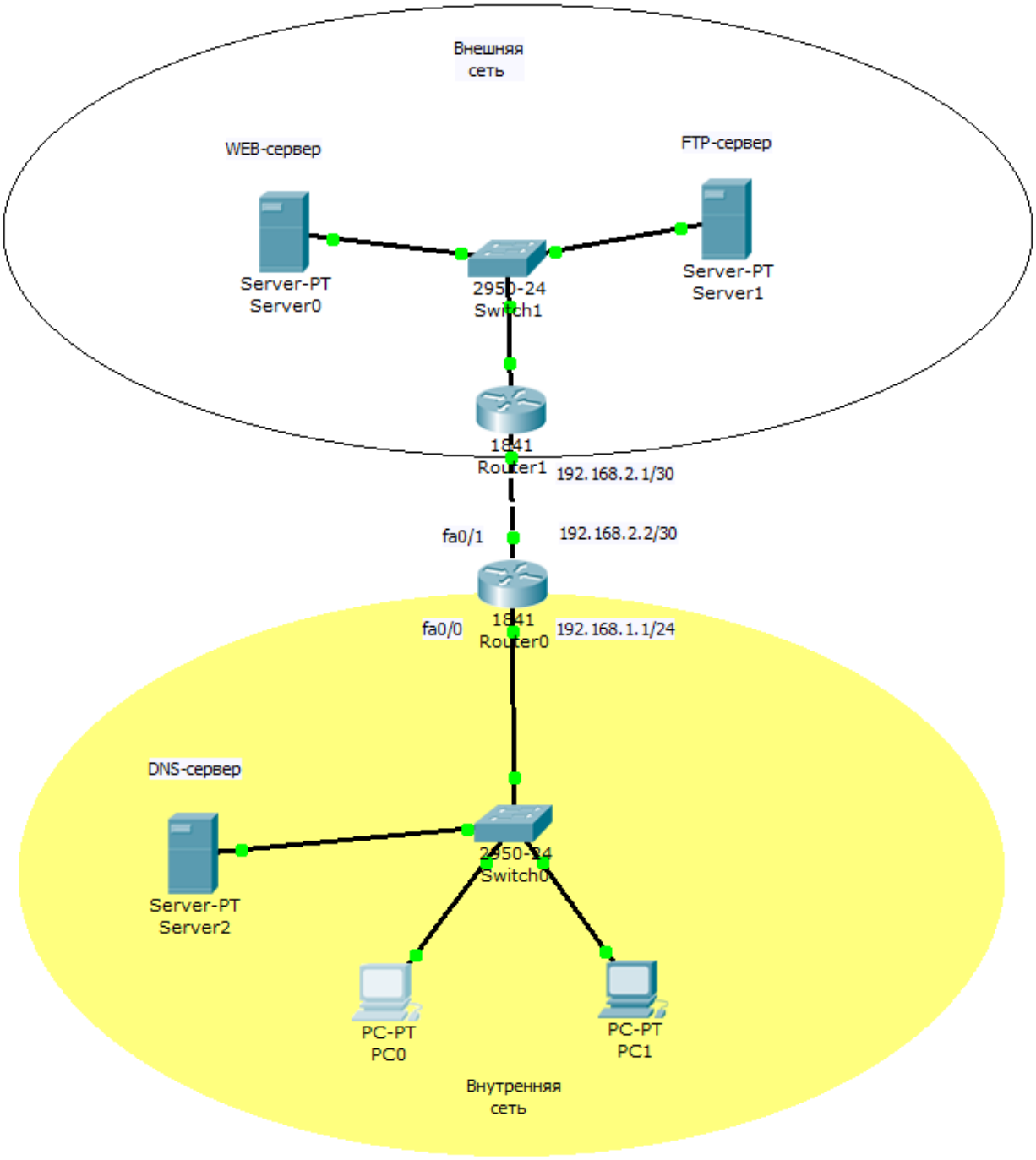

Рисунок 2.1 – Пример сети

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip inspect name HTTP http
Router(config)#int fa0/0
Router(config-if)#ip inspect HTTP in
Router(config-if)#
```

Рисунок 2.2 – 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 2.3).

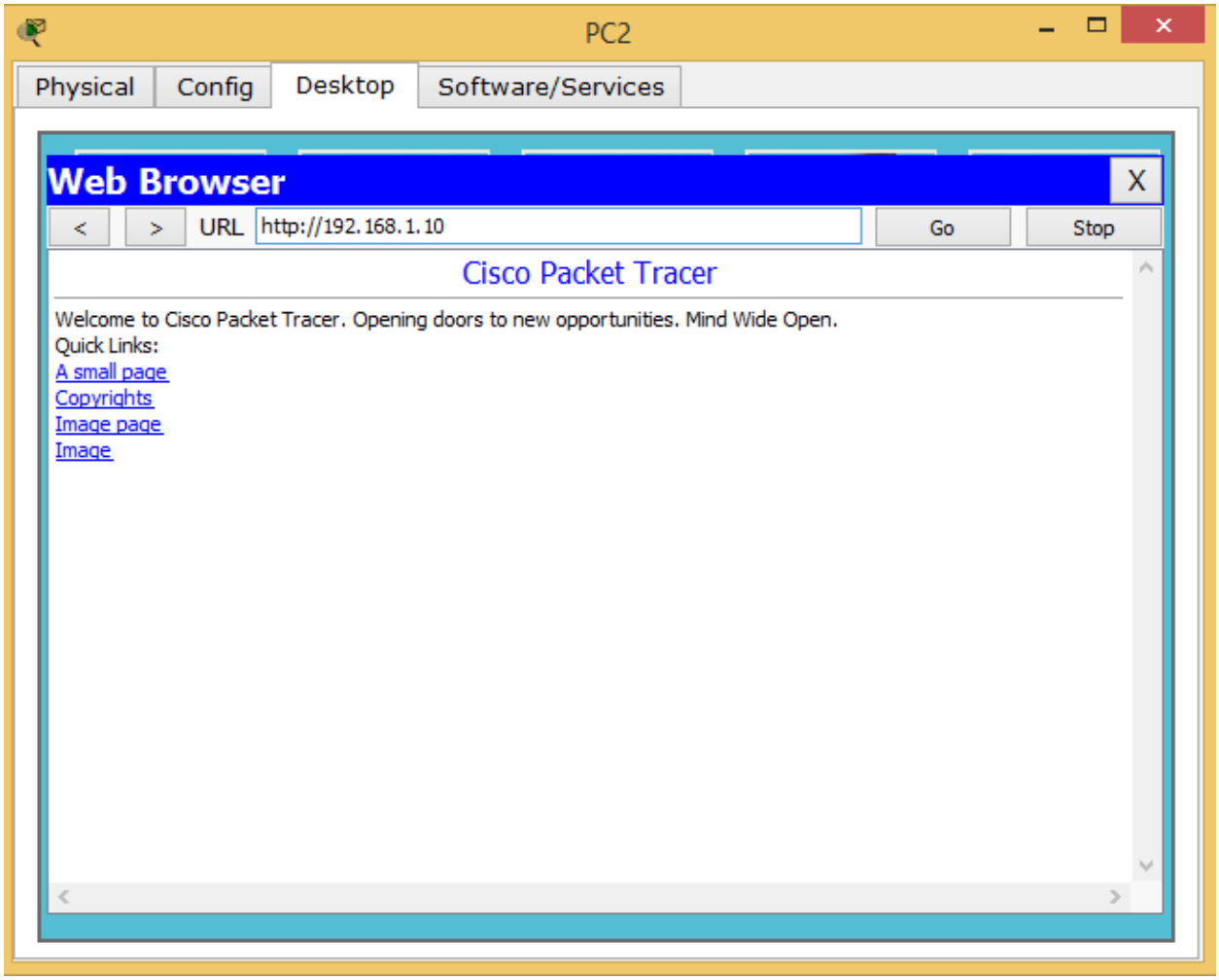

Рисунок 2.3 – Соединение с сервером внутренней сети по протоколу HTTP

В качестве внутреннего сервера мы использовали DNS-сервер с адресом 192.168.1.10/24. Как видно из рисунка 2.3, 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 2.4).

```
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 2.4 – Создание списка доступа

Теперь снова попытаемся послать HTTP-запрос из внешней сети (рисунок 2.5).

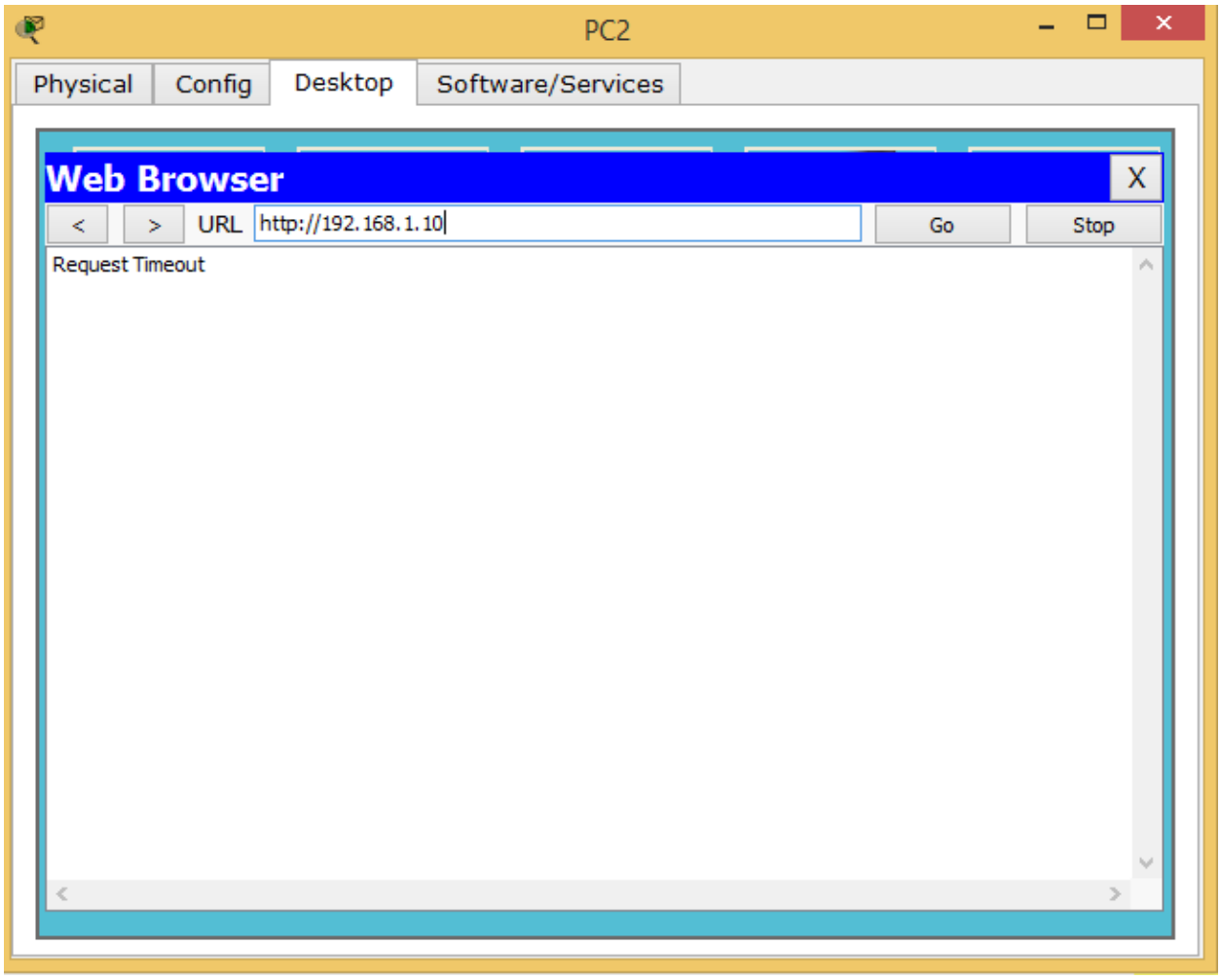

Рисунок 2.5 – Отсутствие соединения с сервером внутренней сети по протоколу HTTP

Как видно из рисунка 2.5, 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 2.6).

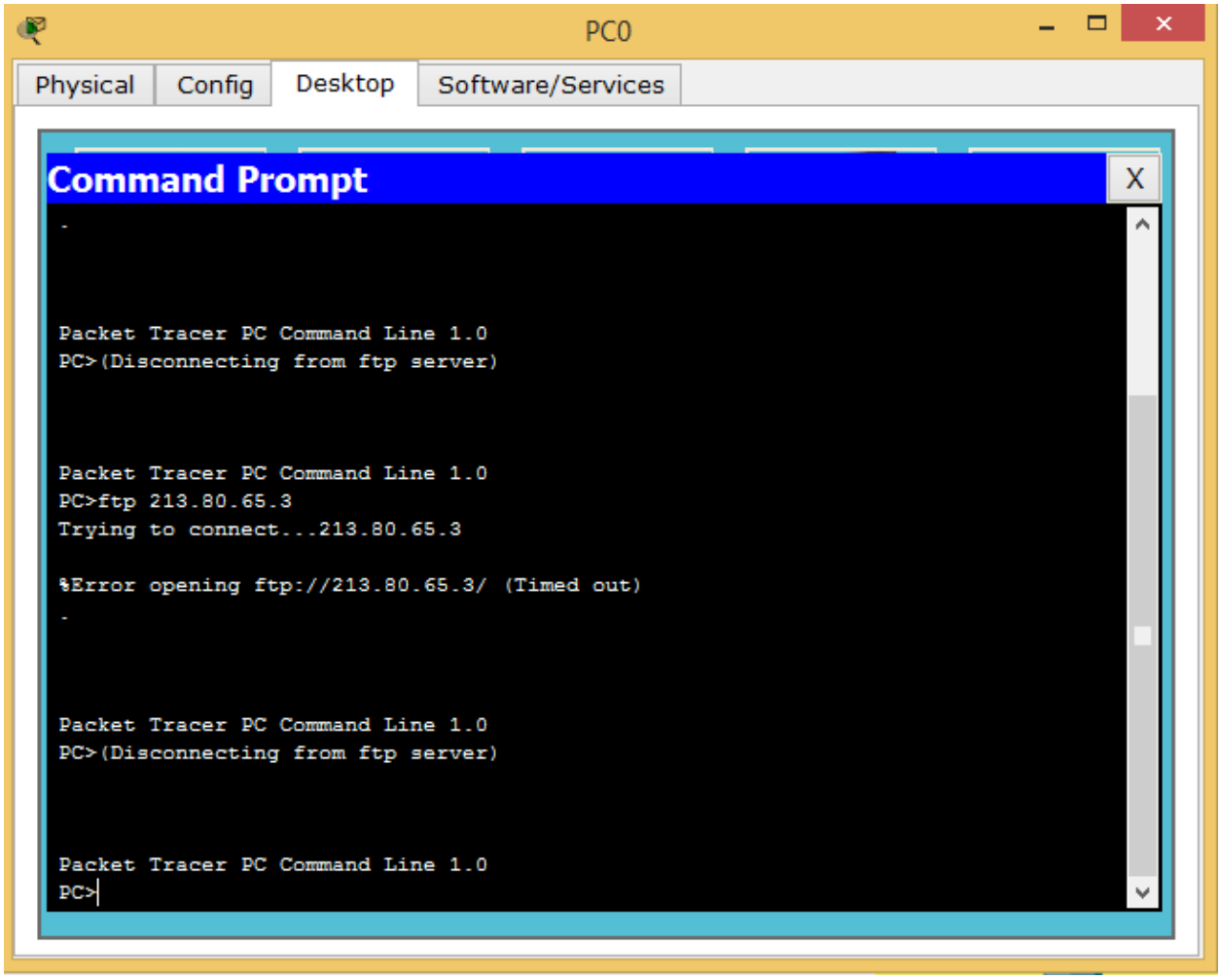

Рисунок 2.6 – Попытка соединения с ftp-сервером

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 2.7, 2.8).

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

Рисунок 2.7 – Настройка инспектирования TCP-трафика

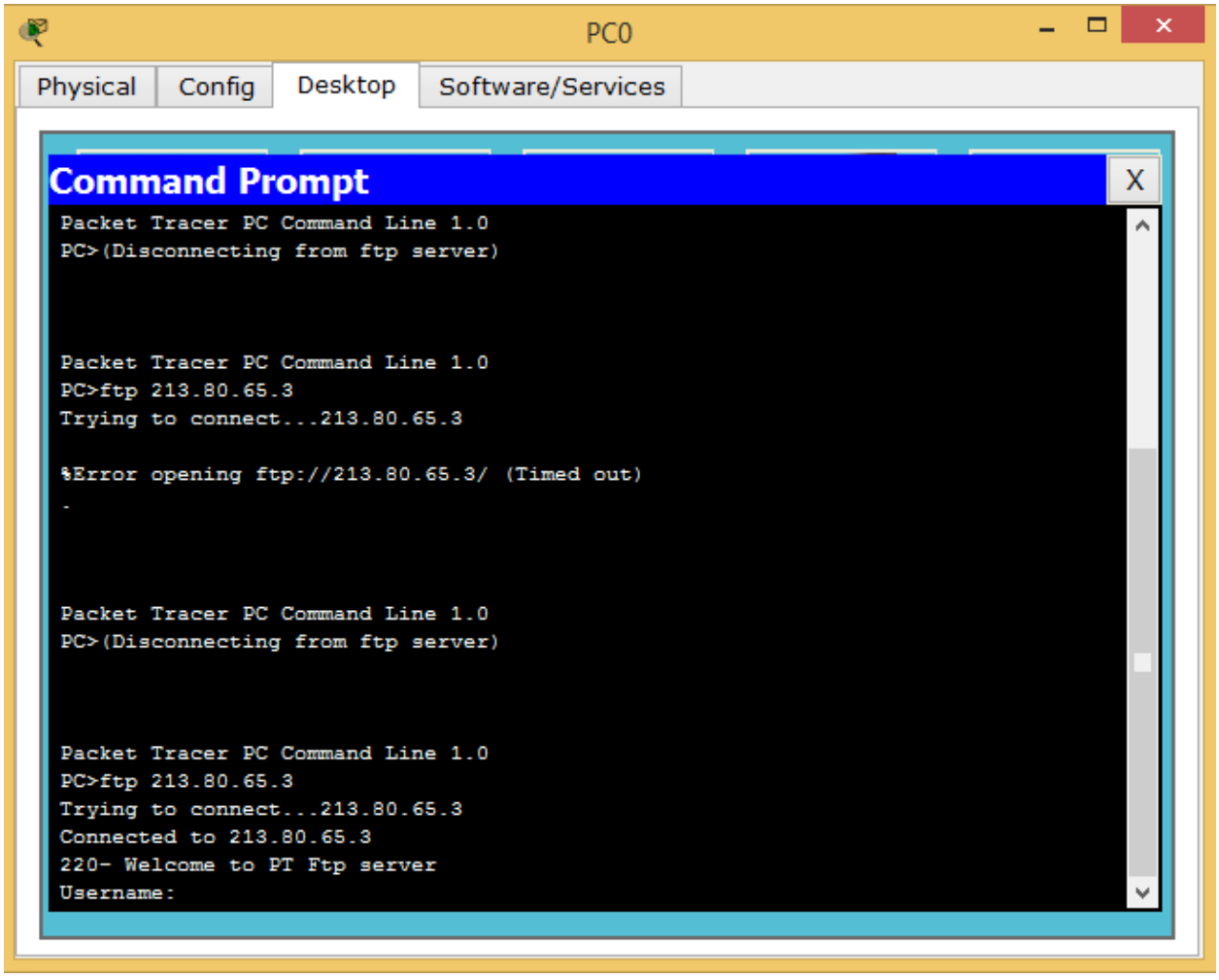

Рисунок 2.8 – Доступ к ftp-серверу после настройки инспектирования TCP-трафика

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

2.5 Отчет по работе:

- Демонстрация функционирования МСЭ.

Лабораторная работа 3. Исследование свойств межсетевого экрана ZBFW

1.1 Цель работы: Получение навыков конфигурирования межсетевого экрана ZBPF и исследование его свойств.

1.2 Перечень оборудования:

- Локальная сеть;
- ПК с установленным ПО Cisco Packet Tracer;
- Маршрутизаторы Cisco.

1.3 Задание:

- Собрать сеть, показанную на рисунке 3.1, 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.
- Проверить связность сети.
- 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.
- Проверить работоспособность межсетевого экрана.

1.4 Указания к проведению работы.

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 (рассмотренных в параграфе 3.1) 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 3.1.

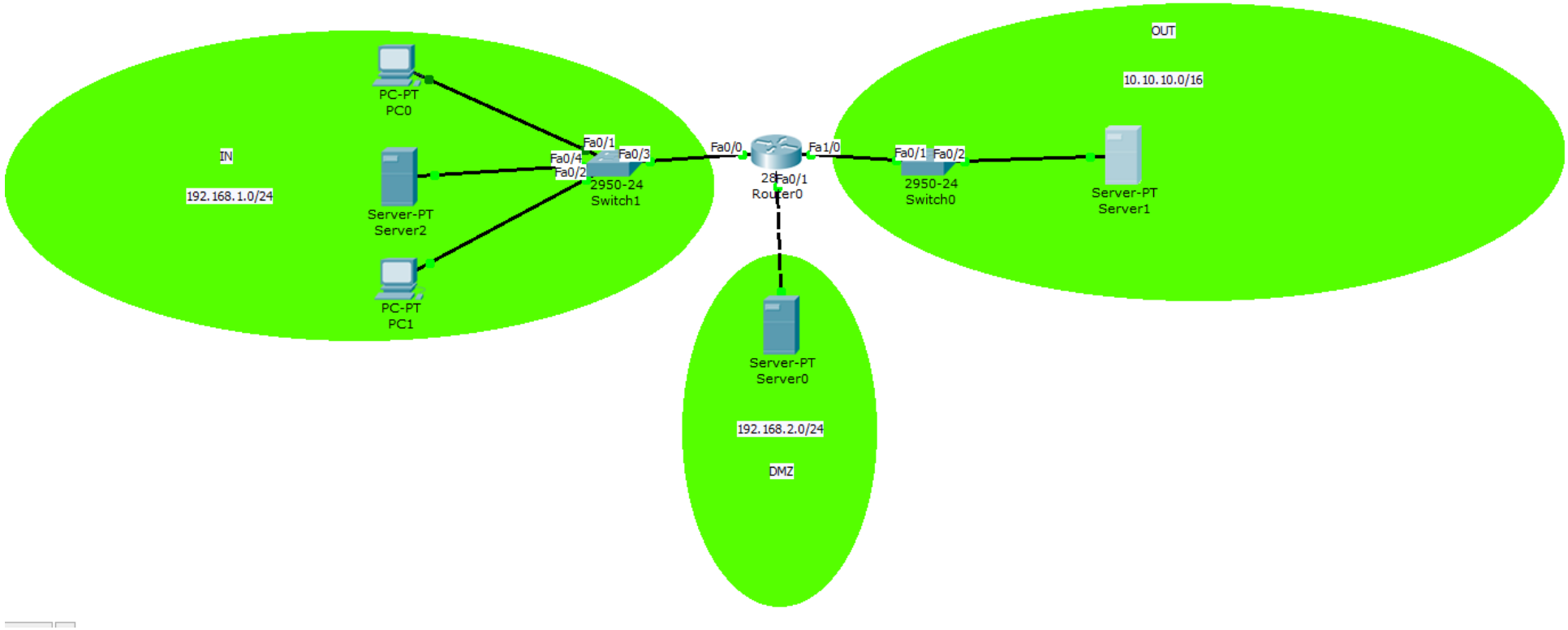

Рисунок 3.1 – Пример сети

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 3.2). Если же послать запрос с внешнего сервера, ответа не будет – запрос будет отброшен маршрутизатором (рисунок 3.3).

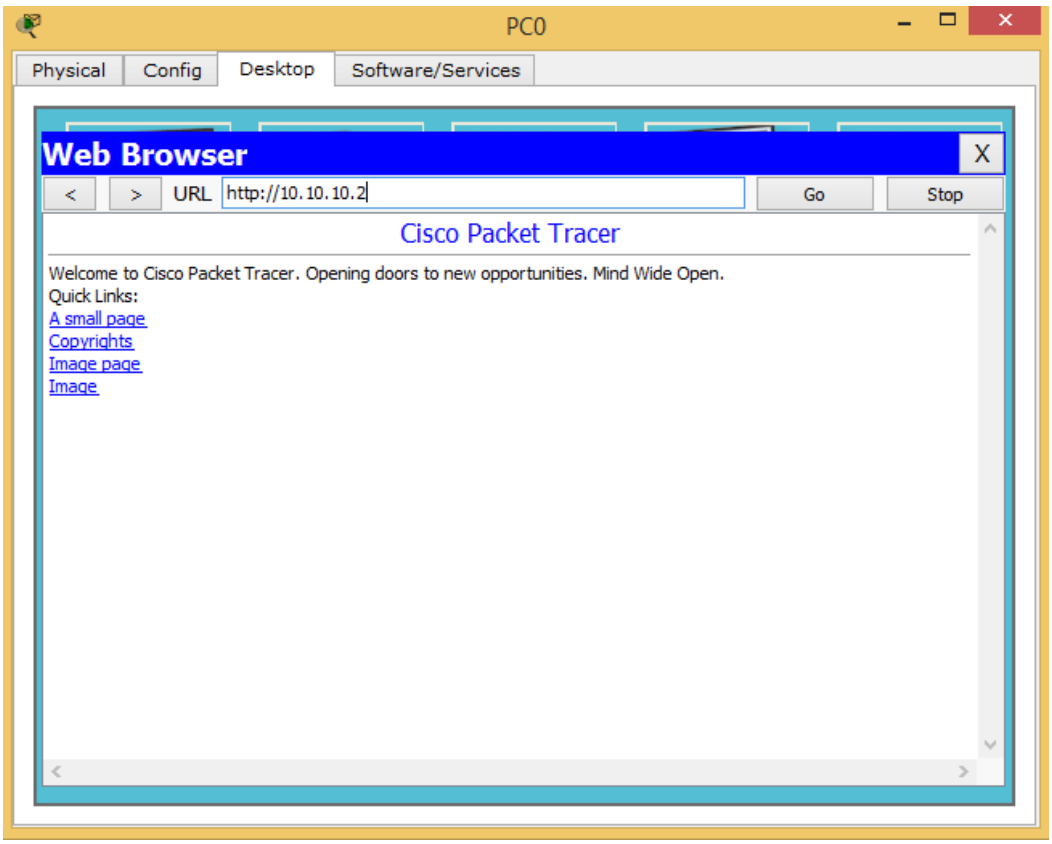

Рисунок 3.2 – Получение ответа при запросе из внутренней сети

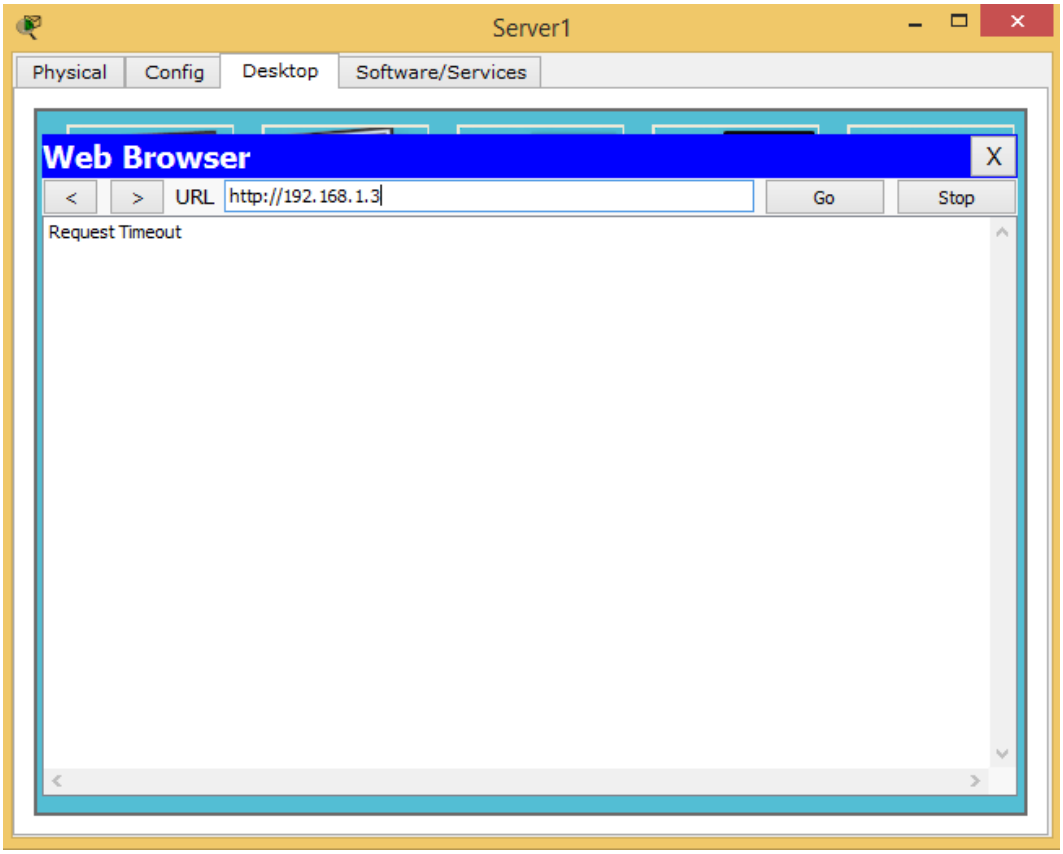

Рисунок 3.3 – Отсутствие ответа при запросе из внешней сети

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

3.5 Отчет по работе:

- Демонстрация функционирования МСЭ.

Лабораторная работа 4. Исследование VPN-туннеля

1.1 Цель работы: Получение навыков конфигурирования VPN-туннеля и исследование его свойств.

1.2 Перечень оборудования:

- Локальная сеть;
- ПК с установленным ПО Cisco Packet Tracer;
- Маршрутизаторы Cisco.

1.3 Задание:

- Собрать сеть, состоящую из двух маршрутизаторов.
- Сконфигурировать VPN-туннель.
- Проверить работоспособность туннеля.

1.4 Указания к проведению работы.

Здесь рассмотрим только 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 или интернет-киоска. 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. 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 4.1.

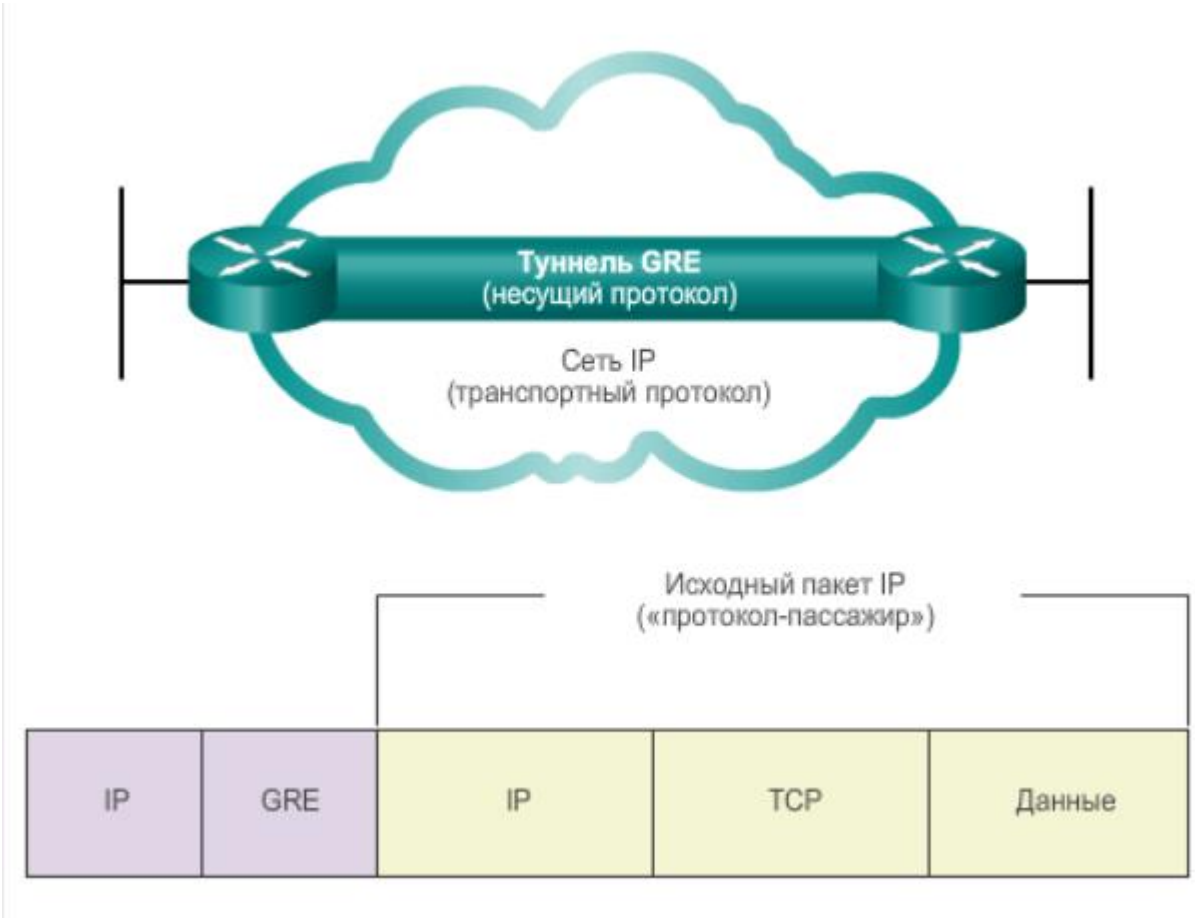

Рисунок 4.1 – Инкапсуляция с использованием GRE

Рассмотрим пример, заимствованный из [8].

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 4.2.

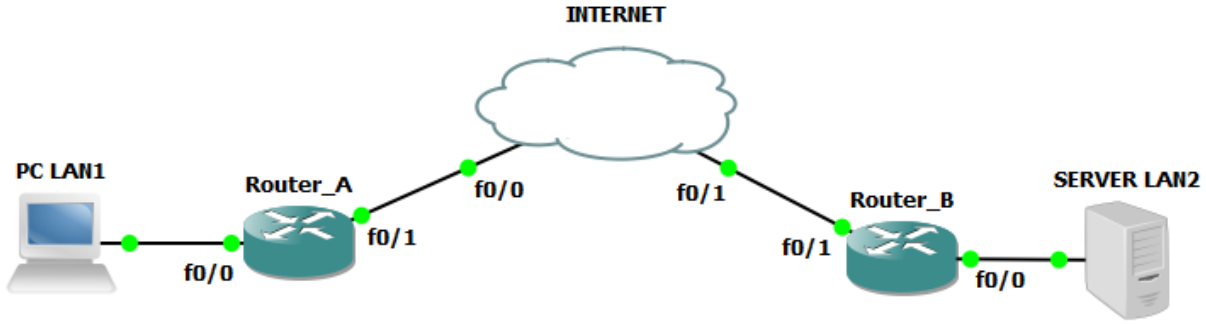

Рисунок 4.2 – Пример объединения двух частных сетей

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 4.3. Анализ содержимого пакета, передаваемого по туннелю, показан на рисунке 4.4.

Из рисунка 4.4 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 4.5.

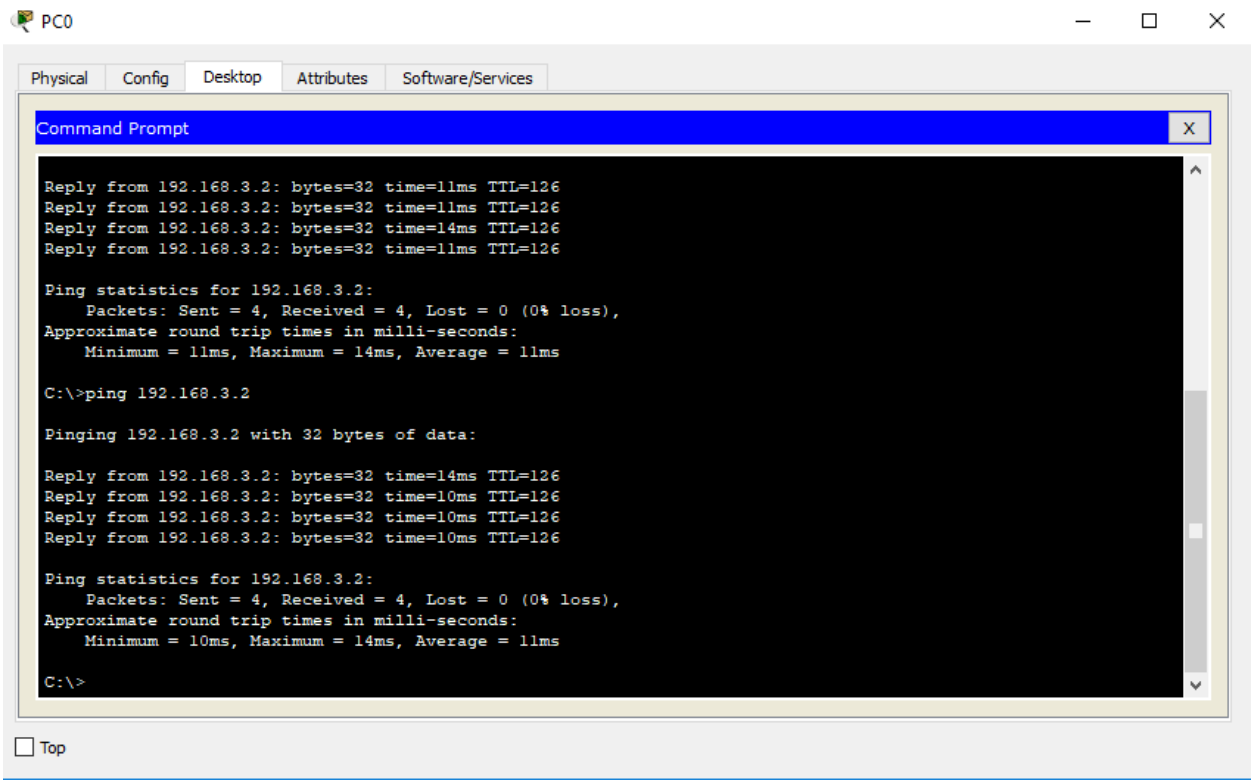

Рисунок 4.3 – Доступ ко второй локальной сети из первой с использованием туннеля

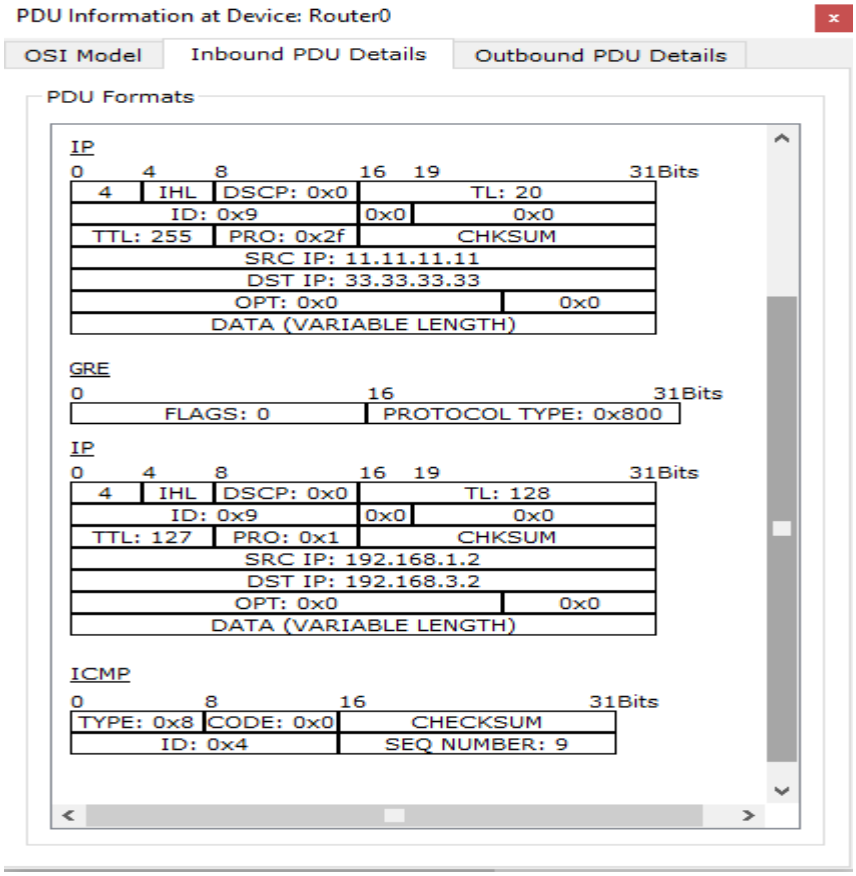

Рисунок 4.4 – Анализ содержимого пакета, передаваемого по туннелю

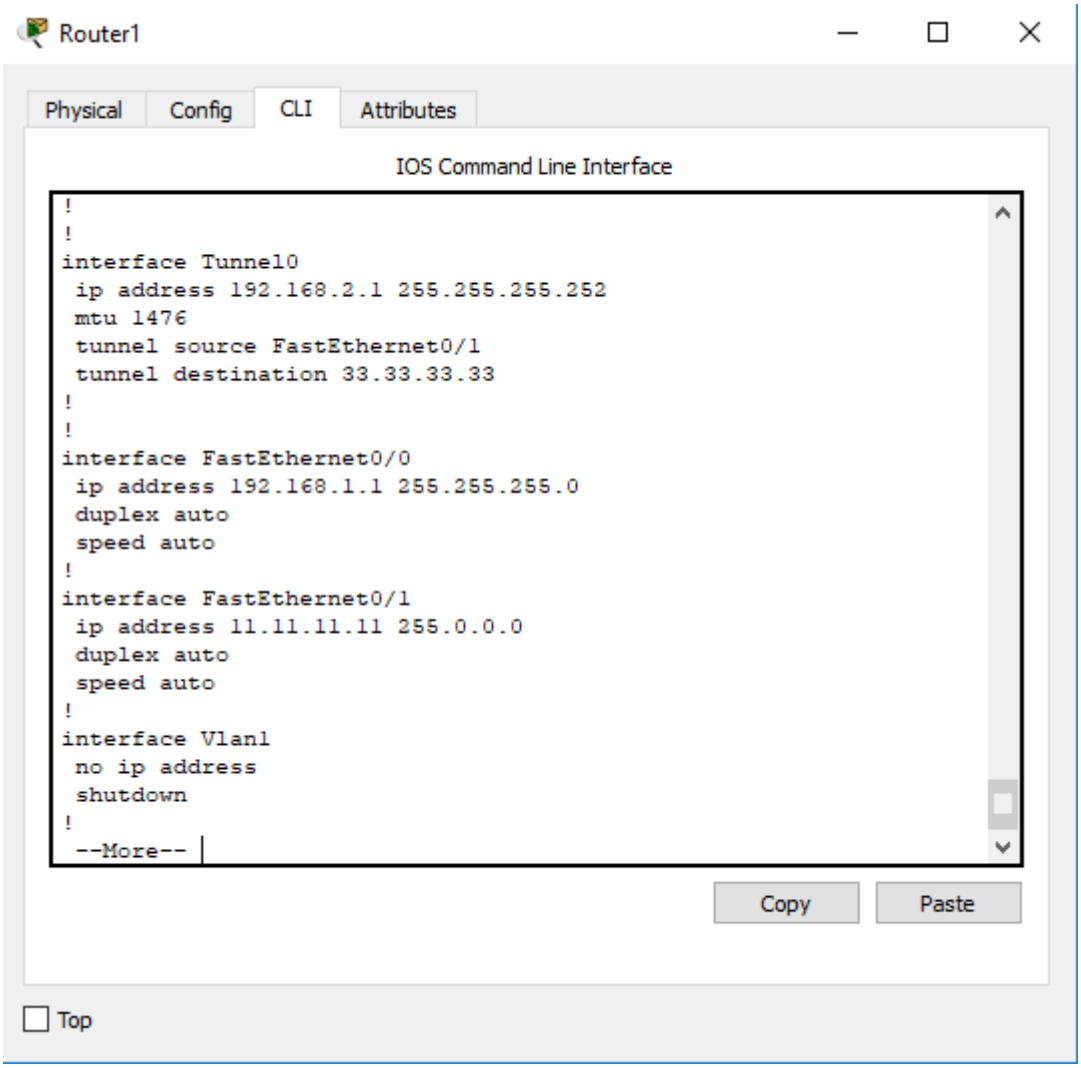

Рисунок 4.5 – Просмотр интерфейса tunnel0 на маршрутизаторе

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

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. IPsec находит применение в организации VPN-соединений.

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 4.6.

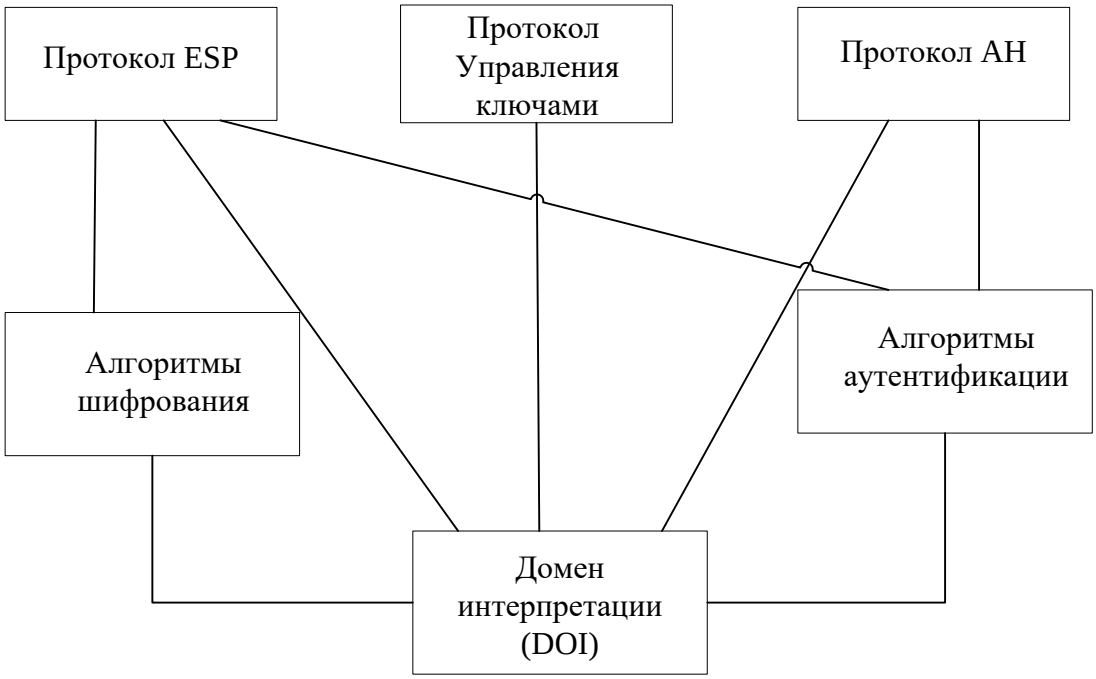

Рисунок 4.6 – Архитектура IPSec

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. 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, 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, 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 4.6 – 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.

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 4.7 показаны исходный пакет, пакеты после обработки протоколом AH в транспортном и туннельном режимах.

Рисунок 4.7 – Форматы пакетов

Формат AH-заголовка показан на рисунке 4.8.

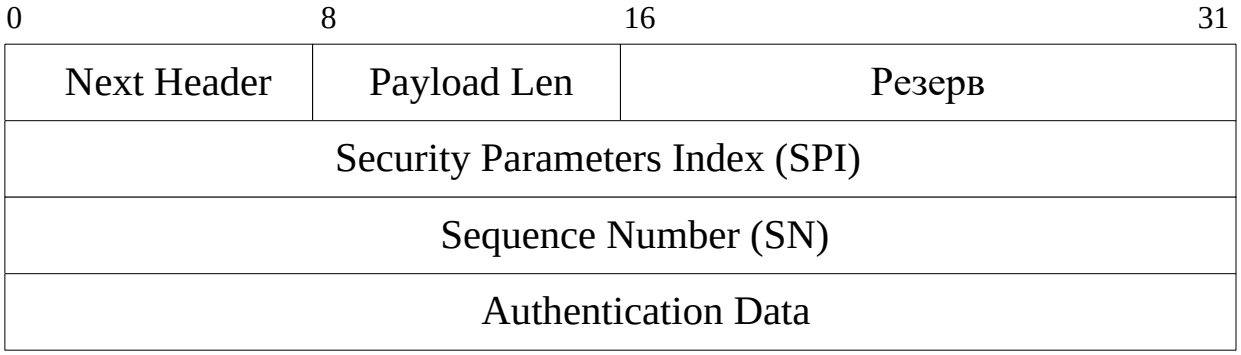

Рисунок 4.8 – Формат заголовка AH

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.

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 4.9. На рисунке серым фоном отмечены поля пакета, подлежащие шифрованию.

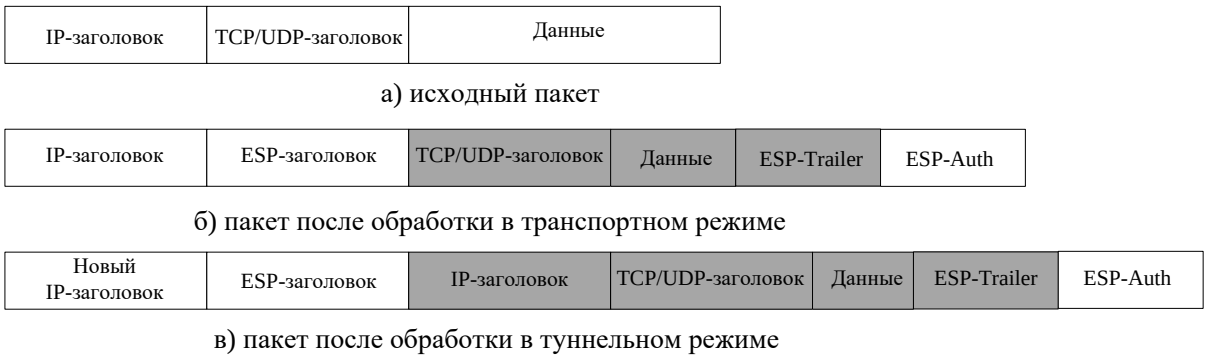

Рисунок 4.9 – Форматы пакетов

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 4.10. Это не столько заголовок, сколько обертка (инкапсулирующая оболочка, как видно из рисунка 4.10) для зашифрованного содержимого. Например, ссылку на следующий заголовок нельзя выносить в начало, в незашифрованную часть, так как она лишится конфиденциальности.

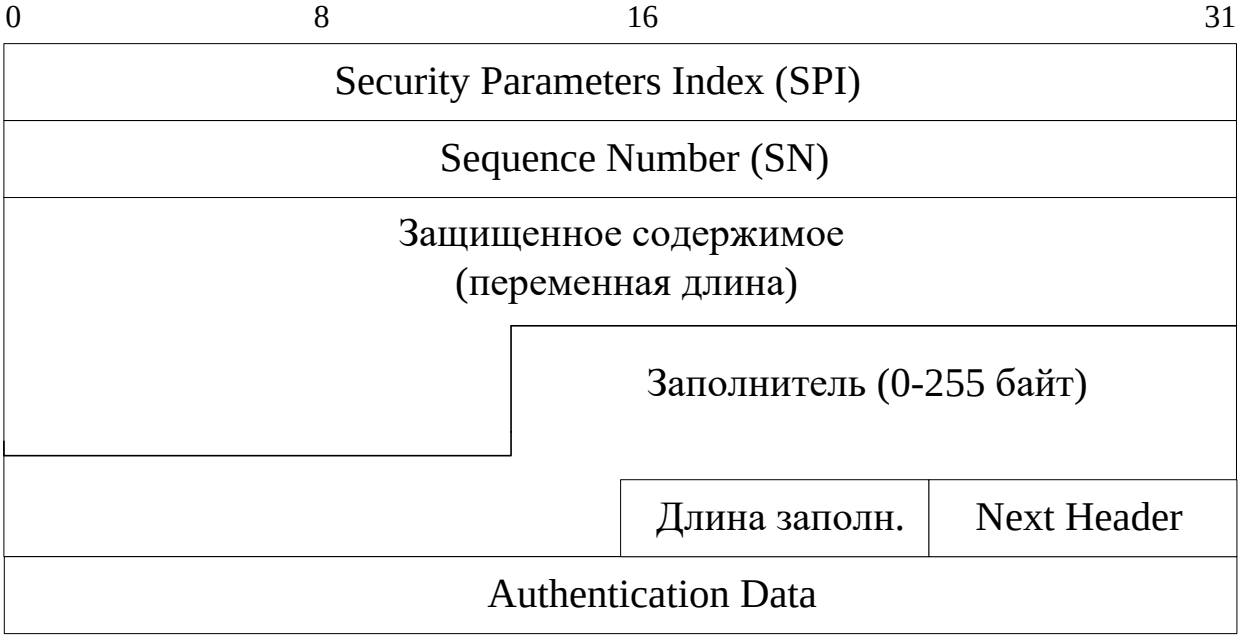

Рисунок 4.10 – Формат заголовка ESP

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 4.9). 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).

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 сначала конфигурирование VPN с туннелированием. Часто такой режим называют IPSec-over-GRE. Будем использовать тот же пример, рисунок 4.11.

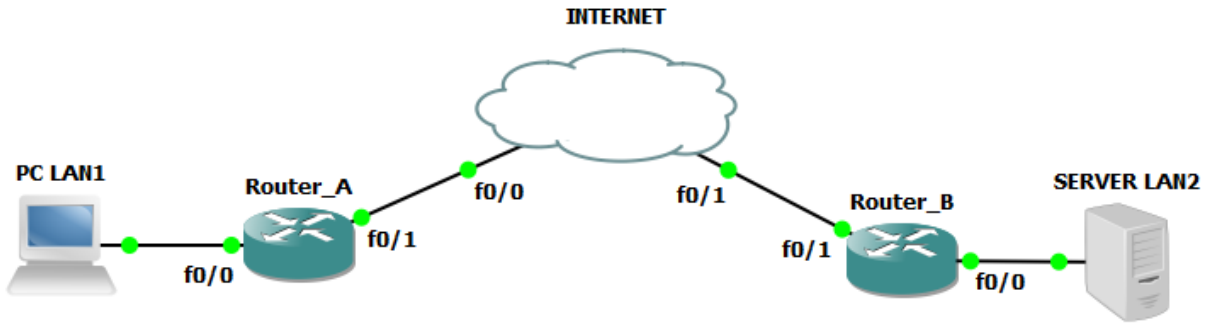

Рисунок 4.11 – Пример объединения двух частных сетей

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 [9]:

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 4.12, заимствованным из [10].

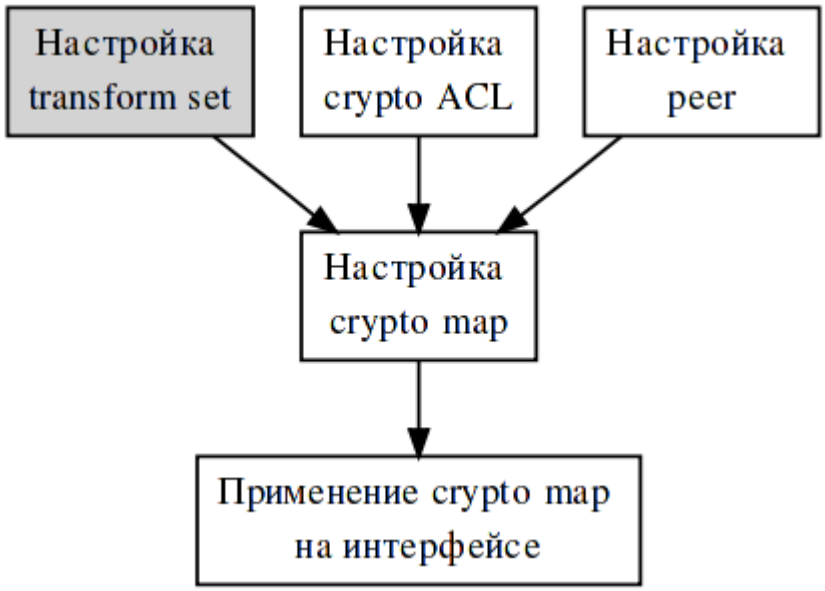

Рисунок 4.12 – Принцип конфигурирования transform-set

Из рисунка 4.12 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 4.1представлены каждый из возможных вариантов.

Таблица 4.1 – 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 4.12). 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

RouterA (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 4.13.

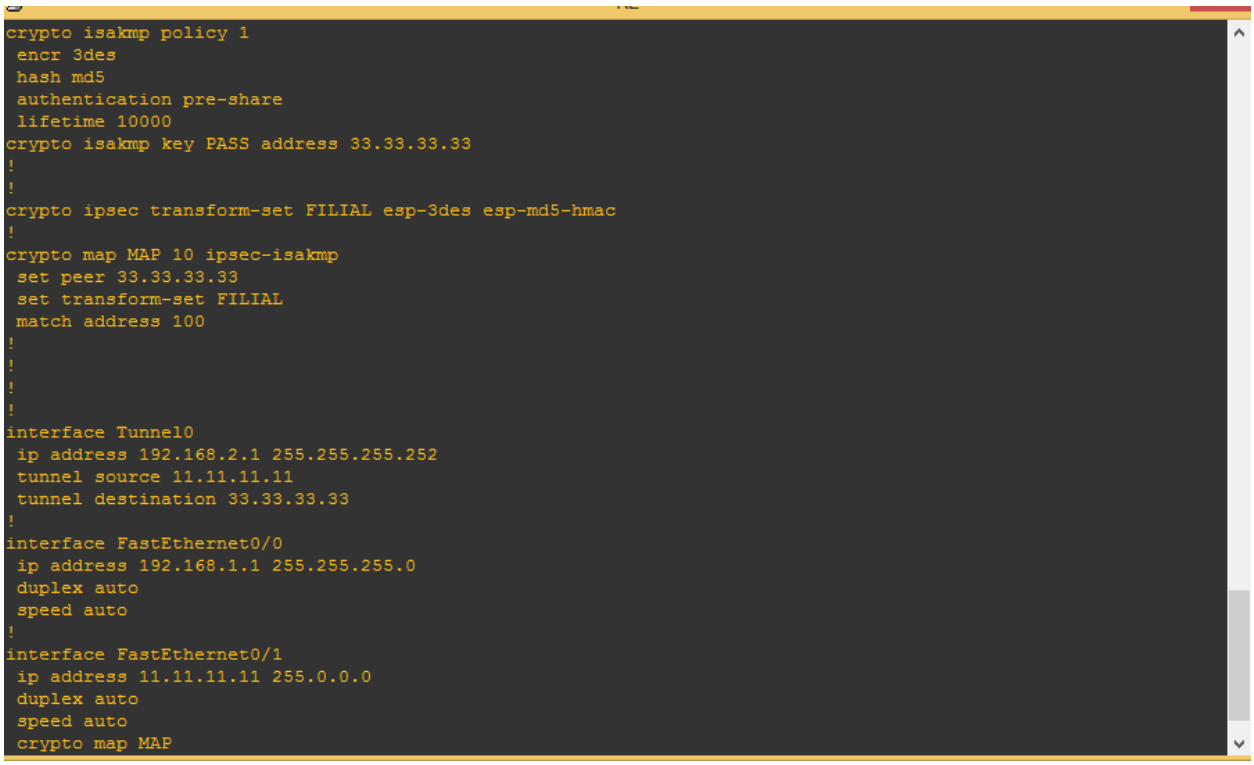

Рисунок 4.13 – Просмотр конфигурации

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 4.14 – 4.17.

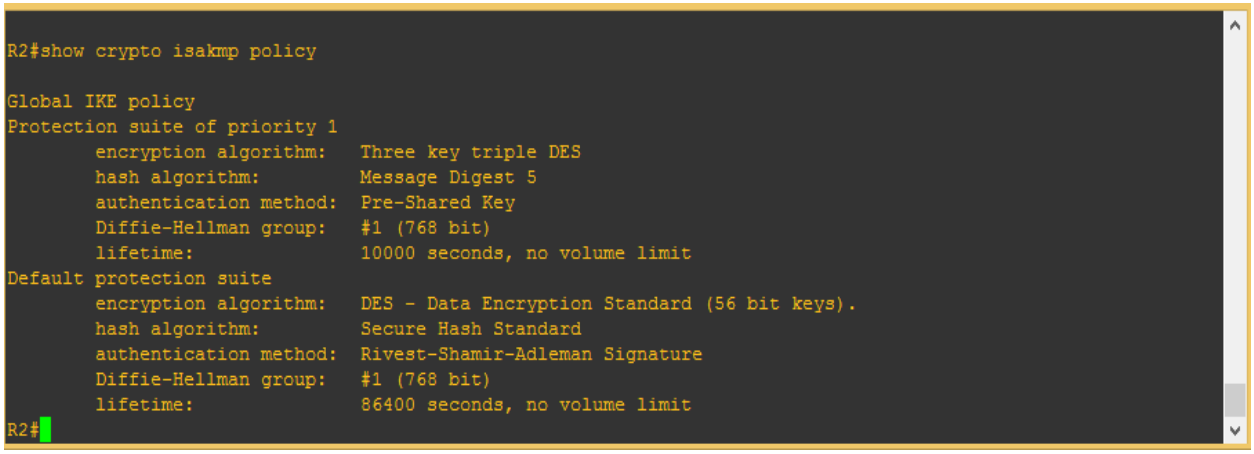

Рисунок 4.14 – Результат выполнения команды **show crypto isakmp policy**

Из рисунка видно, что на маршрутизаторе имеются две политики – одна созданная нами (1), и вторая – по умолчанию.

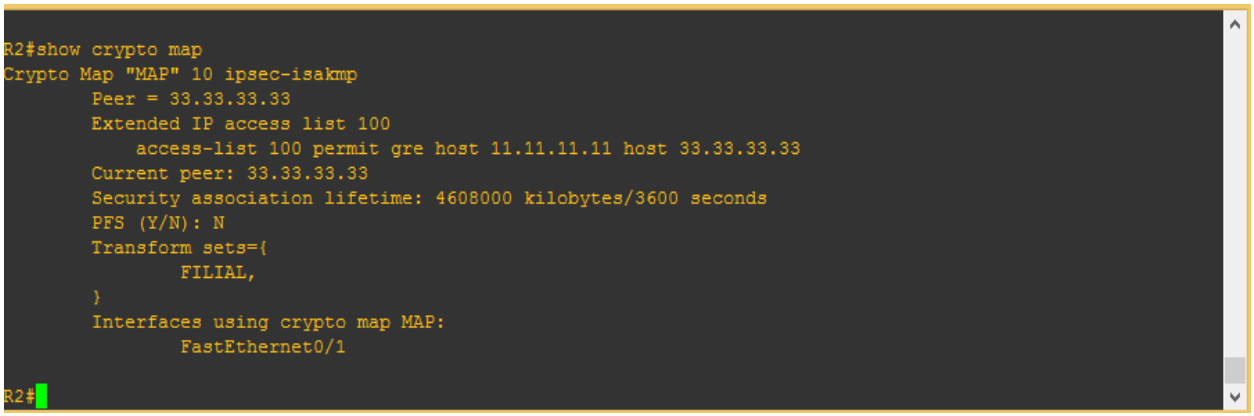

Рисунок 4.15 – Результат выполнения команды **show crypto map**

Рисунок 4.16 – 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**.

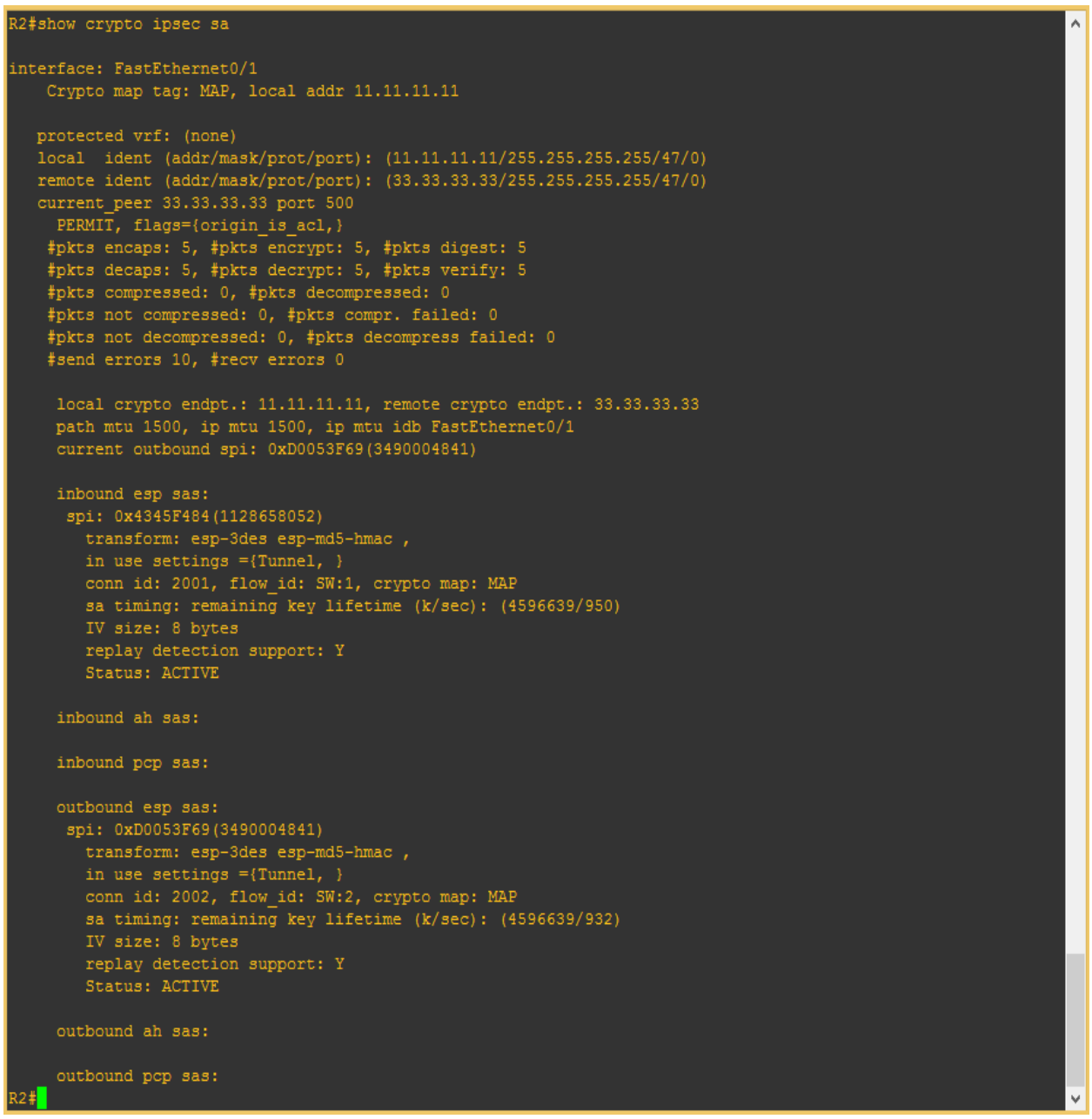

Рисунок 4.17 – Результат выполнения команды **show crypto ipsec sa**

Количество зашифрованных и расшифрованных пакетов можно просмотреть с использованием команды **show crypto engine connections active**, рисунок 4.18.

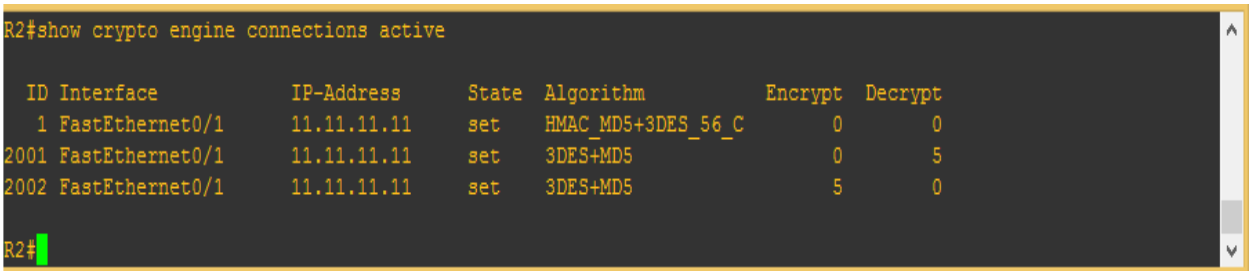

Рисунок 4.18 – Результат выполнения команды **show crypto engine connections active**

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 ACL_CRYPTO

permit ip 192.168.1.0 0.0.0.255 192.168.4.0 0.0.0.255

Кроме того, при настройке IPSec-туннеля можно использовать профили.

4.5 Отчет по работе

Сконфигурированный VPN-туннель на маршрутизаторах Cisco.

Библиографический список:

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: СКФ МТУСИ, 2018.

2. Ожиганов А.А. Криптография. Учебное пособие. СПб: Университет ИТМО, 2016.

3. https://www.intuit.ru/studies/courses/2/2/lecture/50

4. https://www.gns3.com/

5. https://www.wireshark.org/

6. https://tacacsgui.com/

7. Малюха В.А., Новопашенный А.Г., Подгурский Ю.Е., Заборовский В.С. Методы и средства защиты компьютерной информации. Межсетевое экранирование: Учебное пособие. СПб: Изд-во СПбГПУ, 2010.

8. http://yztm.ru/2018/03/18/vpncisco1/

9. https://techprofi.com/network/nastraivaem-vpn-ipsec-cisco/

10. http://xgu.ru/wiki/IPsec_%D0%B2_Cisco