

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Севастопольский государственный университет»

**ИССЛЕДОВАНИЕ СПОСОБОВ КОНФИГУРАЦИИ
СПИСКОВ КОНТРОЛЯ ДОСТУПА
В ЛОКАЛЬНЫХ КОМПЬЮТЕРНЫХ СЕТЯХ**

**Методические указания
к лабораторной работе
по дисциплине**

«Инфокоммуникационные системы и сети»

Для студентов, обучающихся по направлению 09.03.02
«Информационные системы и технологии»
и 09.03.03 «Прикладная информатика»
по учебному плану подготовки бакалавров
дневной и заочной форм обучения

**Севастополь
2024**

УДК 004.732

Исследование способов конфигурации списков контроля доступа в локальных компьютерных сетях. Методические указания к лабораторным занятиям по дисциплине «Инфокоммуникационные системы и сети» / Сост., В.С. Чернега – Севастополь: Изд-во СевГУ, 2024 – 22 с.

Методические указания предназначены для проведения лабораторных работ по дисциплине «Инфокоммуникационные системы и сети». Целью методических указаний является помощь студентам в исследовании способов назначения стандартных и расширенных списков контроля доступа (ACL). Излагаются теоретические и практические сведения необходимые для выполнения лабораторной работы, требования к содержанию отчета.

Методические указания рассмотрены и утверждены на методическом семинаре и заседании кафедры информационных систем

Рецензент: Моисеев Д.В., д.т.н., проф. кафедры ИТиКС

Лабораторная работа

ИССЛЕДОВАНИЕ СПОСОБОВ КОНФИГУРАЦИИ СПИСКОВ КОНТРОЛЯ ДОСТУПА В ЛОКАЛЬНЫХ КОМПЬЮТЕРНЫХ СЕТЯХ

1 ЦЕЛЬ РАБОТЫ

Исследование методов контроля доступа к сетевым ресурсам и способов составления списков ограничения доступа, приобретение практических навыков составления стандартных и расширенных списков доступа, а также конфигурации сетевого оборудования.

2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1 Общая характеристика списков контроля доступа

В процессе реализации политики сетевой безопасности одной из важнейших задач является возможность закрытия доступа для некоторых пакетов. Для отсеечения нежелательных пакетов широко применяются списки доступа *ACL (Access Control Lists)*, которые являются своеобразными фильтрами пакетов. Они позволяют запретить или разрешить определенным хостам доступ к ресурсам сети. Например, в корпоративной сети администраторы могут запретить доступ к внутреннему серверу или в Интернет определенным пользователям, а остальным наоборот – разрешить.

Списки доступа позволяют фильтровать трафик на входе и выходе интерфейсов маршрутизаторов. На входе весь поступающий трафик подвергается фильтрации. Нежелательные пакеты отбрасываются и уже только потом остальные пакеты маршрутизируются. Если же ACL настроены на выходе интерфейса, то трафик фильтруется сразу же после процесса маршрутизации. Списки доступа позволяют использовать маршрутизатор как межсетевой экран (файрвол, брандмауэр) для запрета или ограничения доступа к внутренней сети из внешней, например, сети Интернет. Брандмауэр, как правило, помещается в точках соединения между двумя сетями.

Списки доступа содержат набор инструкций (директив, предписаний) какие порты и адреса блокировать, а какие наоборот разрешить. Этих инструкций в списке может быть от единиц до нескольких десятков. В конце списка всегда содержится неявная инструкция по блокировке всего трафика. Данная инструкция добавляется автоматически самой системой. В настройках она не видна, но нужно учитывать ее наличие.

На настоящее время существуют 3 типа списков доступа: 1) стандартные; 2) расширенные и 3) именованные списки.

Стандартные списки позволяют проверять только IP адрес отправителя. Стандартные списки доступа рекомендуется устанавливать как можно ближе к

отправителю. Стандартные и расширенные списки доступа обязательно нумеруются. Номера стандартных списков могут принимать значение от 0 до 99.

Расширенные списки управления доступом *extended ACL* (*extended Access Control List*) используются чаще, чем стандартные, поскольку они обеспечивают большие возможности контроля. Расширенные списки предписывают проверку как адреса отправителя, так и адреса получателя. Они могут также проверять конкретные протоколы, номера портов и другие параметры. Это придает им большую гибкость в задании проверяемых условий. Пакету может быть разрешена отправка или отказано в передаче в зависимости от того, откуда он был выслан и куда направлен. Расширенным спискам доступа назначаются номера от 100 до 199.

Именованные списки аналогичны стандартным и расширенным ACL, но вместо номеров используются имена списков. Стандартные и расширенные списки редактировать нельзя. К примеру, нельзя в середину списка вставить команду или удалить ее. Для этого нужно сначала **деактивировать список на самом интерфейсе**, а затем полностью его удалить и настроить заново. Именованные списки позволяют редактировать вновь созданные списки. Все введенные команды нумеруются, что позволяет легко добавлять и удалять команды.

Динамические списки ACL дают системным администраторам возможности предоставить пользователю временный доступ или ограничить доступ к маршрутизатору из интернета. Динамические списки используют для расширенных ACL, Telnet и аутентификации.

Настройка динамических списков контроля доступа начинается с применения расширенного списка ACL для блокирования трафика через маршрутизатор. Пользователи, пытающиеся передать данные через маршрутизатор, блокируются расширенным списком ACL до тех пор, пока они не войдут на маршрутизатор через Telnet и не будут аутентифицированы. Затем подключение Telnet сбрасывается, а динамический список ACL с одной записью добавляется к существующему расширенному списку ACL. Таким образом разрешается трафик на определенный период времени; можно задавать таймаут простоя и абсолютный таймаут.

2.2 Правила составления списков доступа

Правила построения и назначения списков доступа для различных протоколов имеют свою специфику, однако, можно выделить два этапа работы с любыми списками доступа. Сначала, создается список доступа, а затем выполняется привязка его к соответствующему интерфейсу, линии связи или логической операции, выполняемой маршрутизатором (роутером).

Каждое предписание в списке доступа записывается отдельной строкой. Список доступа в целом представляет собой набор строк с директивами, имеющих один и тот же номер (или имя). Порядок задания директив в списке играет важную роль. Проверка пакета на соответствие списку производится последовательным применением предписаний из данного списка (в том порядке, в

котором они были внесены). В конце каждого списка системой IOS добавляется неявное правило, состоящее в том, что если пакет удовлетворяет какому-либо предписанию, то дальнейшие проверки его на соответствие следующим директивам в списке НЕ ПРОИЗВОДЯТСЯ. Таким образом, пакет, который не соответствует ни одному из введенных предписаний, отвергается.

Для одного списка можно определить несколько директив. Каждая из них должна ссылаться на имя или на номер списка для того, чтобы все они были связаны с одним и тем же списком. Количество директив может быть произвольным, и ограничено лишь объемом имеющейся памяти. Однако, чем больше в списке директив, тем труднее понять логику работы списка и контролировать ее. Поэтому рекомендуется тщательно заносить всю информацию о списках в специальный журнал.

Как уже упоминалось выше, порядок строк в списке доступа очень важен, поскольку невозможно изменить этот порядок или исключить какие-либо строки из существующего списка доступа. По этой причине целесообразно предварительно создавать списки доступа (например, на tftp-сервере) и загружать их целиком в маршрутизатор, а не пытаться редактировать их на маршрутизаторе. Если список доступа с данным номером (именем) существует, то строки списка с тем же номером (именем) будут добавляться к существующему списку в конец его.

Списки управления доступом представляют собой перечень особых **директив** (предписаний): «разрешить» (*permit*) и «запретить» (*deny*). Эти директивы применяются к адресам или протоколам верхних уровней (3-7). Предписание «разрешить» означает, что все пакеты, отвечающие определенным условиям, будут пропущены, т.е. им будет разрешено дальнейшее перемещение по сети. Директива «запретить» указывает, что пакет, имеющий определенные характеристики, необходимо удалить. Списки доступа могут применяться для запрещения продвижения пакетов через определенный интерфейс маршрутизатора в ту или другую сторону, для ограничения доступа некоторых пользователей и устройств к сетевым ресурсам, для указания способа шифрования, а также для указания приоритетности обработки пакетов.

Каждая из директив в списке доступа читается процессором маршрутизатора по порядку, т.е. очередной пакет, проходящий через соответствующий порт, будет последовательно сравниваться со всеми критериями (адресом источника, адресом получателя или номером порта) **в списке доступа с начала списка до конца**. Если пакет не соответствует условию первой директивы, то он проверяется на соответствие второй директиве из списка управления доступом. А если параметры пакета соответствуют следующему условию, которое представляет собой директиву разрешения доступа, то ему разрешается отправка на интерфейс получателя. Таким образом, при первом обнаружении соответствия остальные директивы не рассматриваются. Поэтому, если была записана директива, разрешающая передачу всех данных, то все последующие директивы не проверяются. Следует особо подчеркнуть, что **в конце каждого списка выполняется неявное правило "deny all"** (запретить все), поэтому при назначении списков на интерфейс нужно следить, чтобы явно разрешить все виды

необходимого трафика через интерфейс (не только пользовательского, но и служебного, например, обмен информацией по протоколам динамической маршрутизации).

Для создания **стандартного списка доступа** для маршрутизаторов Cisco применяется команда **access-list**, которая вводится в следующем формате:

access-list <номер_списка> {deny|permit} <адрес отправителя> [маска шаблона адреса] [log]

Маска шаблона (англ. *wildcard mask*) указывает маршрутизатору на те биты в шаблоне адреса отправителя, которые следует сравнивать с поступившим в порт маршрутизатора адресом отправителя, и те, которые нужно проигнорировать. Как и маска в схеме адресации протокола IP, маска шаблона в списке доступа состоит из 32-х битов, записанных в точечно-десятичной форме. Например, маска шаблона 0.0.0.255 соответствует двоичному представлению 00000000.00000000.00000000.11111111. Однако **запись маски шаблона** в списках доступа, в отличие от метода записи маски адреса на сетевых интерфейсах, **записана инверсно**, т.е. единицами отмечены биты адреса, которые НЕ будут проверяться. **Нулевые биты** в маске списка доступа предписывают маршрутизатору **необходимость сравнения** соответствующих битов IP-адреса в проверяемом пакете с аналогичными битами в шаблоне адреса. Соответственно, единичные биты указывают на то, что сравнение производить не нужно. Таким образом, маска 0.0.0.0 вынуждает маршрутизатор сравнивать все 32 бита адреса пакета на соответствие их битам шаблона, заданного в списке доступа.

Ключевое слово **"log"** инициирует выдачу записи о совпадении пакета с данным предписанием на консоль и в системный лог-файл. Часто используемое описание фильтра, которому удовлетворяет любой адрес 0.0.0.0 255.255.255.255, имеет специальное обозначение **"any"**:

access-list <access-list-number> {deny | permit} any

В **расширенном списке** перед полем адреса источника можно указывать **тип протокола**, а после адреса источника указываются (при необходимости) адрес хоста назначения и порт. Общий формат расширенного списка доступа имеет следующий вид:

access-list номер-списка {permit | deny} {**протокол**} {адрес источника} [маска-источника][адрес получателя] [маска-получателя] [оператор номер порта] [established] [log]

Параметры списка могут принимать значения:

оператор — lt, gt, eq, neq (меньше, чем, больше, чем, равно, не равно);
established — разрешает прохождение ответного TCP-потока (от web-сервера) во внутреннюю сеть, если он использует установленное из внутренней сети соединение (т. е. если бит ACK в заголовке сегмента установлен). Частные случаи списков доступа могут иметь вид:

access-list access-list-number {deny|permit} **протокол** any any

или

access-list access-list-number {deny | permit} протокол host source host destination

Если в качестве протокола указано "tcp" или "udp", то описания source- и destination-wildcard могут включать номера портов для данных протоколов с ключевыми словами "eq" (*equal*) — равно, "neq" (*not equal*) — не равно, "lt" (*less than*) — меньше чем, "gt" (*greater then*) — больше чем, "range" — указание диапазона номеров портов. Для протокола "tcp", возможно также применение слова "established" для выделения только установленных tcp-сессий. Ключевое слово "host source" эквивалентно записи: "source 0.0.0.0".

При использовании именованных списков в список добавляется оператор, указывающий на стандартный (*standard*) или расширенный (*extended*) список доступа. Синтаксис такого списка имеет вид:

ip access-list {standard | extended} <имя ACL>

Пример формирования стандартного списка.

Запретить прохождение через маршрутизатор пакетов с рабочей станции с IP-адресом 235.12.60.23 и пропустить все остальные.

В связи с тем, что запрет осуществляется только по адресу источника, используем стандартный список доступа, присвоив ему номер 4.

access-list 4 deny host 235.12.60.23

access-list 4 permit any

При составлении сценариев конфигурации маршрутизаторов следует помнить, что команды всех видов списков доступа вводятся в режиме конфигурирования маршрутизатора, который индицируется промптом: Router(config)#.

Для того чтобы список доступа начал выполнять свою функцию, он должен быть применен к интерфейсу с помощью команды

Router(config-if)#ip access-group номер-списка-доступа in|out

Список доступа может быть применен как входной (in) либо как выходной (out). Когда список доступа применяется как входной, то маршрутизатор получает входной пакет и сверяет содержащийся в нем адрес с элементами списка. Маршрутизатор разрешает пакету маршрутизироваться на интерфейс назначения, если пакет удовлетворяет разрешающим элементам списка, либо отбрасывает пакет, если он соответствует условиям запрещающих элементов списка. Если список доступа применяется как выходной, то маршрутизатор получает входной пакет, пересылает его на интерфейс назначения и только тогда проверяет содержащийся в пакете адрес согласно элементам списка доступа этого интерфейса. Далее маршрутизатор либо разрешает отправку пакета через выходной интерфейс, либо отбрасывает его согласно разрешающим и запрещающим директивам списка соответственно. Так, созданный ранее список, например, с номером 77 применяется к интерфейсу Ethernet 0 маршрутизатора как входной список следующими командами:

Router(config)#int Ethernet 0

Router(config-if)#ip access-group 77 in

Этот же список применяется к интерфейсу Ethernet 0 маршрутизатора как выходной список с помощью команд

Router(config-if)#ip access-group 77 out

Отменить список на интерфейсе можно с помощью команды **no**

Router(config-if)#no ip access-group 77 out

Рассмотрим принцип создания более сложных списков доступа. Пусть имеем сеть, изображенную на рисунке 2.1.

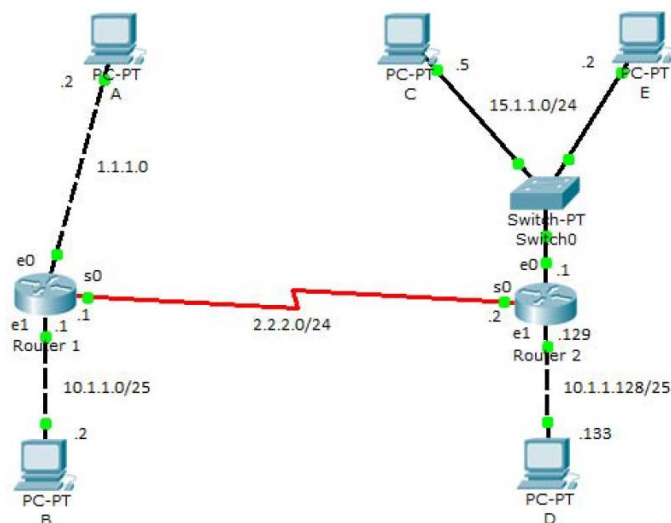


Рисунок 2.1 – Пример топологии сети для создания сложных списков доступа

Разрешим все пакеты, исходящие из сети 10.1.1.0/25 (10.1.1.0 255.255.255.128), но запретим все пакеты, поступающие из сети 10.1.1.128 /25 (10.1.1.128 255.255.255.128) по последовательному интерфейсу S0. Пусть также необходимо запретить все пакеты, исходящие из сети 15.1.1.0 /24 (15.1.1.0 255.255.255.0), за исключением пакетов от единственного хоста с адресом 15.1.1.5. Все остальные пакеты разрешаем. Списку присвоим номер 2. Последовательность команд для выполнения поставленной задачи будет следующая:

```
Router(config)#access-list 2 deny 10.1.1.128 0.0.0.127
Router(config)#access-list 2 permit 15.1.1.5 0.0.0.0
Router(config)#access-list 2 deny 15.1.1.0 0.0.0.255
Router(config)#access-list 2 permit 0.0.0.0 255.255.255.255
```

Отметим отсутствие разрешающего элемента для сети 10.1.1.0 255.255.255.128. Его роль выполняет последний элемент **access-list 2 permit 0.0.0.0 255.255.255.255**.

Удостоверимся, что поставленная задача выполнена.

1. Разрешить все пакеты, исходящие из сети 10.1.1.0 255.255.255.128. Последняя строка в списке доступа удовлетворяет этому критерию. Нет необходимости в явном виде разрешать эту сеть в нашем списке доступа так, как в

списке нет строк, соответствующей этой сети за исключением последней разрешающей строки `permit 0.0.0.0 255.255.255.255`.

2. Запретить все пакеты, исходящие из сети 10.1.1.128 255.255.255.128. Первая строка в списке выполняет этот критерий. Инверсная маска 0.0.0.127 для этой сети предписывает, что не нужно брать в рассмотрение последние семь бит четвертого октета адреса, которые назначены для адресации компьютера в данной подсети. Маска для этой сети 255.255.255.128, которая указывает, что последние семь бит четвертого октета определяют адресацию компьютера в данной сети.

3. Запретить все пакеты, исходящие из сети 15.1.1.0 255.255.255.0, за исключением пакетов от единственного хоста с адресом 15.1.1.5. Это требование удовлетворяется второй и третьей строкой нашего списка доступа. Важно отметить, что список доступа осуществляет это требование не в том порядке как оно определено. Обязательно следует помнить, что список доступа обрабатывается сверху вниз и **при первом совпадении обработка пакетов прекращается**. Поэтому вначале запрещаются все пакеты, исходящие из сети 15.1.1.0 255.255.255.0 и лишь затем разрешаются пакеты с адресом 15.1.1.5. Если в командах, определяющих список доступа, переставить вторую и третью команды, то вся сеть 15.1.1.0 будет запрещена до разрешения хоста 15.1.1.5. То есть, адрес 15.1.1.5 сразу же в начале будет запрещен более общим критерием **deny 15.1.1.0 0.0.0.255**.

4. Разрешить все остальные пакеты. Последняя команда разрешает все адреса, которые не соответствуют первым трем командам.

Таким образом, последовательность действий для реализации списка доступа может быть записана в следующем виде.

1. Определить критерии и ограничения для доступа.

2. Реализовать их с помощью команд `access-list`, создав список доступа с определенным номером.

3. Применить список к определенному интерфейсу либо как входящий, либо как исходящий.

Следует заметить, что в общем случае стандартный список доступа нужно помещать как можно ближе к точке назначения, а не к источнику пакетов. Если список помещен вблизи источника пакетов, то очень вероятно, что доступ к устройствам, на которых не осуществляется никакая конфигурация доступа, будет затруднен.

Конкретизируем политику безопасности для сети на рисунке 2.1. Наша цель создать политику для компьютера А (адрес 1.1.1.2 сеть 1.1.1.0/24), которая из всех устройств локальной сети 15.1.1.0/24, в которую входит компьютер С (15.1.1.5), разрешит доступ к компьютеру А только компьютеру С. Требуется также создать политику, запрещающую удаленный доступ к компьютеру А из любого устройства локальной сети 10.1.1.128 / 25 компьютера D (10.1.1.133). Весь остальной трафик мы разрешаем. На рисунке 2.1 компьютер PC5 (15.1.1.5) играет роль произвольного, отличного от компьютера С, представителя локальной сети 15.1.1.0/24.

Размещение списка критично для реализации такой политики. Возьмем созданный ранее список с номером 2. Если список сделать выходным на последо-

вательном интерфейсе S0 маршрутизатора 2, то задача для компьютера А будет выполнена, однако возникнут ограничения на трафик между другими локальными сетями. Аналогичную ситуацию получим, если сделаем этот список входным на последовательном интерфейсе маршрутизатора 1. Если мы поместим этот список как выходной на интерфейс Ethernet0 маршрутизатора 1, то задача будет выполнена безо всяких побочных эффектов.

Именованные ACL

К именованным ACL обращаются по имени, а не по номеру, что дает наглядность и удобство для обращения. Для создания именованного ACL имеется команда

```
Router(config)#ip access-list standard|extended ACL_name
```

и далее команды для создания элементов списка

```
Router(config-ext-nacl)#permit|deny IP_protocol source_  
IP_address  
wildcard_mask [protocol_information] destination_IP_address  
wildcard_mask [protocol_information] [log]
```

Для завершения создания списка следует дать команду exit.

Имя именованного списка чувствительно к регистру. Команды для создания неименованного списка аналогичные командам для создания элементов нумерованного списка, но сам процесс создания отличен. Вы должны использовать ключевое слово ip перед главным ACL оператором и тем самым войти в режим конфигурации именно для этого именованного списка. В этом режиме вы начинаете с ключевых слов permit или deny и не должны вводить access-list в начале каждой строки.

Привязка именованных ACL к интерфейсу осуществляется командой

```
Router(config)#interface type [slot_№] port_№  
Router(config-if)#ip access-group ACL_name in|out
```

Именованный ACLs разрешает себя редактировать. Для этого надо набрать команду, которая была использована для его создания

```
Router(config)#ip access-list standard|extended ACL_name
```

С помощью клавиш с вертикальными стрелками следует найти строку списка, которую нужно изменить. Изменить ее, используя горизонтальные стрелки. Нажать ввод. Новая строка добавится в конец списка. Старая не уничтожится. Для ее удаления следует ввести **no** в начале строки.

Для редактирования же числовых ACLs необходимо его уничтожить и создать заново или изменить список онлайн и загрузить в устройство с помощью telnet.

Пример именованного списка доступа

Создается стандартный список доступа с именем Internet_filter и расширенный список доступа с именем marketing_group:

```

Router(config)#interface Ethernet0/5
Router(config-if)#ip address 2.0.5.1 255.255.255.0
Router(config)#ip access-group Internet_filter out
Router(config-if)#ip access-group marketing_group in
Router(config)#ip access-list standard Internet_filter
Router(config-ext-nacl)#permit 1.2.3.4
Router(config-ext-nacl)#deny any
Router(config)#ip access-list extended marketing_group
Router(config-ext-nacl)#permit tcp any 171.69.0.0 0.0.255.255 eq telnet
Router(config-ext-nacl)#deny tcp any any
Router(config-ext-nacl)#permit icmp any any
Router(config-ext-nacl)#deny udp any 171.69.0.0 0.0.255.255 lt 1024
Router(config-ext-nacl)#deny ip any any log

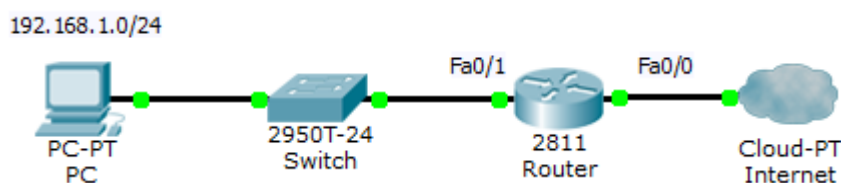
```

Reflexive ACL – зеркальные списки контроля доступа

Зеркальные (reflexive) ACL – это расширение технологии extended ACL, которое позволяет организовать пропуск трафика из Интернета в локальную сеть только в ответ на предварительно сделанный запрос из локальной сети в Интернет. Зеркальные списки ACL можно задать **только** с помощью расширенных именованных списков ACL для протокола IP. Их нельзя определить с помощью нумерованных или стандартных списков ACL для протокола IP или с помощью списков ACL для других протоколов.

Суть фильтрации пакетов состоит в следующем: на выходной интерфейс локальной сети прикрепляется ACL, который пропускает исходящий трафик. Одновременно автоматически формируется встречный ACL для пропуска входящего трафика. Благодаря этому разрешается получать ответы из Интернета только на свои запросы. Ключевые слова, используемые в Reflexive ACL – это **reflect** для исходящего трафика и **evaluate** для входящего.

Пример. Пусть имеется локальная компьютерная сеть с адресом 192.168.1.0/24. Из нее нужно организовать доступ в Интернет всем клиентским компьютерам сети по протоколам http, pop и smtp.



Вначале необходимо создать именованный расширенный список для исходящего трафика, например, с именем IN-TO-OUT, а затем для входящего трафика список именем OUT-TO-IN. При этом следует учитывать, что сообщения для указанных сервисов (http, pop и smtp) передаются по протоколу TCP. Сценарий, реализующий заданные условия, имеет следующий вид.

```

R1(config)#ip access-list extended IN-TO-OUT
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq www reflect BACK-WWW
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq pop3 reflect BACK-POP
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq smtp reflect BACK-SMTP

```

```

R1(config-ext-nacl)#exit
R1(config)#ip access-list extended OUT-TO-IN
R1(config-ext-nacl)#evaluate BACK-WWW
R1(config-ext-nacl)#evaluate BACK-POP
R1(config-ext-nacl)#evaluate BACK-SMTP

```

Здесь BACK-WWW, BACK-POP и BACK-SMTP – имена зеркальных списков доступа. Параметр **reflect name** используется для создания рефлексивного списка доступа.

Затем эти списки связываются с внешним fa0/0 и внутренним fa0/1 интерфейсами маршрутизатора.

```

R1(config)#interface fa0/0
R1(config-if)#ip access-group OUT-TO-IN in
R1(config)#interface fa0/1
R1(config-if)#ip access-group IN-TO-OUT out
R1(config-if)#

```

Список IN-TO-OUT разрешает выход трафика изнутри наружу. Пропускается трафик на порты 25, 80 и 110, параллельно формируются зеркальные ACL с именами BACK-WWW, BACK-POP и BACK-SMTP, которые пропускают обратный трафик. Весь трафик извне фильтруется ACL с именем OUT-TO-IN, который по умолчанию ничего не пропускает, но когда появляются зеркальные записи, то трафик начинает пропускаться.

Предположим, что пользователь обращается с адреса 192.168.1.100 к веб-страничке на сервере 123.123.123.123 при обращении выбирается случайный порт отправителя (например, 1235), порт получателя используется стандартный – 80. Когда пакет проходит через маршрутизатор, он проверяется IN-TO-OUT и на основании первой строчки списка выводится в канал. Одновременно в ACL с именем BACK-WWW автоматически на время добавляется зеркальная запись:

```

permit tcp host 123.123.123.123 eq 80 host 192.168.1.100 eq 1235

```

То есть, в настоящий момент весь трафик из интернета вовнутрь будет заблокирован, за исключением ответа от веб-сервера на наш запрос. Преимущество Reflexive ACL перед established заключается в том, что при established используется только флагом в TCP сегменте, а Reflexive реально отслеживает соединения. Флаг можно подделать, в этом случае входящий трафик начнет пропускаться. Конечно, его вряд ли кто-то примет, но можно устроить, например, DOS атаку. Но самое важное преимущество, с помощью established в принципе нельзя организовать пропуск протоколов, отличных от TCP. Например, протоколов, базирующихся на UDP, или ICMP трафик. Зеркальные же ACL успешно справляются с этими задачами.

ACL обрабатываются сверху вниз. Наиболее часто повторяющийся трафик должен быть обработан в начале списка. Как только обрабатываемый список пакет удовлетворяет элементу списка, обработка этого пакета прекращается. Стандартные ACLs следует помещать ближе к точке назначения, где трафик

должен фильтроваться. Выходные (out) расширенные ACLs следует помещать как можно ближе к источнику фильтруемых пакетов, а входные следует помещать ближе к точке назначения, где трафик должен фильтроваться.

Ограничение доступа виртуального терминала к VTY при помощи ACL

ACL можно применять не только для фильтрации трафика, но и для ограничения адресов, с которых можно подключиться с помощью виртуального терминала к маршрутизатору по telnet или ssh.

Сначала создается стандартный ACL, в котором перечисляются адреса и сети, из которых доступ по telnet надо разрешить. Теперь его необходимо применить непосредственно на **line vty 0 4**, то есть, на линии виртуального терминала, к которым происходит подключение. Таким образом, не важно, через какой интерфейс маршрутизатора telnet-пакеты попадут на роутер, они будут отфильтрованы когда доберутся собственно до vty.

На маршрутизаторе создается стандартный список доступа **VTY_ACCESS**:

```
Router(config)#ip access-list standard VTY_ACCESS
Router(config-std-nacl)#permit 15.15.1.0 0.0.0.255
```

Устанавливается ограничение доступа к **VTY** на маршрутизаторе:

```
Router(config)#line vty 0 4
Router(config-line)#access-class VTY_ACCESS in
```

Теперь по telnet можно подключиться только из сети 15.15.1.0.

Обратите внимание, что ACL применяется на интерфейсе командой **access-group**, а на vty – командой **access-class**.

Пример конфигурации динамического списка ACL.

```
username test password 0 test
username test autocommand access-enable host timeout 10
interface fa0/0
ip address 10.1.1.1 255.255.255.0
ip access-group 101 in
access-list 101 permit tcp any host 10.1.1.1 eq telnet
access-list 101 dynamic testlist timeout 15 permit ip 10.1.1.0 0.0.0.255
172.16.1.0 0.0.0.255
line vty 0 4
login local
```

После того, как пользователь 10.1.1.2 создает подключение Telnet к 10.1.1.1, применяется динамический список ACL. Затем соединение отключается и пользователь может переходить в сеть 172.16.1.x.

3. ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ

В качестве лабораторной установки используется персональный компьютер с установленной программой Packet Tracer, позволяющей осуществлять моделирование компьютерных сетей, построенных на оборудовании корпорации Cisco. Подробно описание пакета моделирования и работы с ним приведено в лабораторной работе №1. Исследуемая схема компьютерной сети изображена на рисунке 3.1.

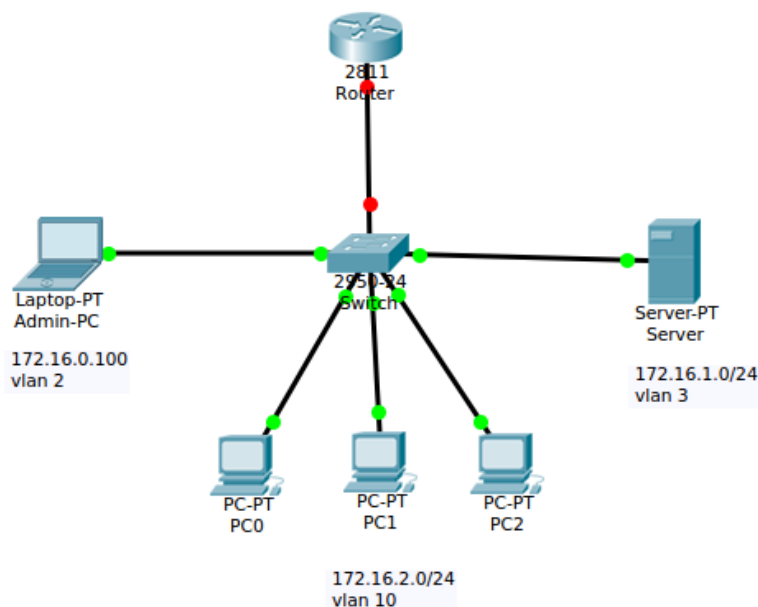


Рисунок 3.1 – Схема исследуемой компьютерной сети

4. Программа выполнения работы

4.1. Изучить теоретический материал, относящийся к составлению и применению списков доступа (выполняется в процессе домашней подготовки).

4.2. Создать в рабочем окне Packet Tracer схему сети, изображенную на рисунке 3.1.

4.3. Сконфигурировать коммутатор таким образом, чтобы компьютер администратора с адресом 172.16.0.100 находился в vlan 2, сервер с адресом 172.16.1.0/24 размещался в vlan 3, а рабочие станции представляли собой подсеть vlan 10 с адресом 172.16.2.0/24. Конфигурацию оборудования выполнить с командной строки.

4.4. Сконфигурировать оборудования таким образом, чтобы доступ к серверу имел только администратор.

4.5. Проверить путем пингования, что требования, изложенные в п.4.3 и 4.4 выполнены.

4.6. Переконфигурировать оборудования таким образом, чтобы пользователи рабочих станций PC0-PC2 имели доступ к файл-серверу и к HTTP (порт 80) и FTP (порт 21) серверам. При этом предусмотреть функционирование DNS (порт 53) сервера.

4.7. Сформулировать выводы по результатам исследований.

Примечание: проверить правильность конфигурации телекоммуникационного оборудования и обнаружить ошибки конфигурации можно путем использования приложения А.

5. Содержание отчета

Отчет о выполненной работе должен содержать:

1. Титульный лист.
2. Схему исследуемой сети и программу работы.
3. Скрипты настроек сетевого оборудования.
4. Скриншоты результатов исследования функционирования сети.
5. Выводы.

6 Контрольные вопросы

- 6.1. Что представляют собой списки контроля доступа?
- 6.2. Какой адрес является критерием для разрешения/запрещения пакета?
- 6.3. Где применяются ACL?
- 6.4. Как задать элемент ACL и что такое инверсная маска?
- 6.5. Как маршрутизатор обрабатывает элементы ACL?
- 6.6. Какой элемент всегда неявно присутствует в ACL?
- 6.7. Как ACL применить к интерфейсу и затем его отменить?
- 6.8. Чем отличается входной ACL от выходного?
- 6.9. Где в сети рекомендуется размещать ACL?
- 6.10. Какими тремя командами можно проверить содержимое ACL и привязку к интерфейсу.
- 6.11. Что фильтруют расширенные ACL?
- 6.12. Какую дополнительную функциональность имеют расширенные ACL по сравнению со стандартными?
- 6.13. Можно ли, используя расширенные ACL, наложить ограничения на трафик к определённой TCP/IP службе?
- 6.14. Опишите процедуру создания именованного ACL.
- 6.15. Как отредактировать конкретную строку в числовом ACL?
- 6.16. Как отредактировать конкретную строку в именованном ACL?
- 6.17. Чем отличаются форматы команд для ввода элементов числового и именованного ACL?
- 6.18. В чем состоит особенность динамических списков доступа?

Библиографический список

1. Дибров, М. В. Сети и телекоммуникации. Маршрутизация в IP-сетях: учебник и практикум для вузов / М. В. Дибров. — 2-е изд., перераб. и доп. — Москва: Юрайт, 2024. — 423 с. — URL: <https://urait.ru/bcode/544928> (дата обращения: 19.04.2024).
2. Сети и телекоммуникации: учебник и практикум для вузов / К. Е. Самуйлов [и др.] ; под редакцией К. Е. Самуйлова, И. А. Шалимова, Д. С. Кулябова. — 2-е изд., перераб. и доп. — Москва: Юрайт, 2024. — 464 с. — URL: <https://urait.ru/bcode/536089> (дата обращения: 17.04.2024).
3. Урбанович, П. П. Компьютерные сети: учебное пособие / П. П. Урбанович, Д. М. Романенко. - Москва; Вологда: Инфра-Инженерия, 2022. — 460 с. — URL: <https://znanium.com/catalog/product/1902692> (дата обращения: 17.04.2024).
4. Чернега В.С. Компьютерные сети / В.С. Чернега, Б. Платтнер. — Севастополь: Изд-во СевНТУ, 2006. — 500 с.

Приложение А. Сценарий реализации пунктов программы 4.3 и 4.4.
[<https://pcradar.ru/nastroyka-acl-v-cisco/>]

```
Switch>enable - переходим в расширенный режим
Switch#configure terminal - переходим в режим конфигурации
Switch(config)#vlan 2 - создаем vlan 2
Switch(config-vlan)#name Admin - название для vlan 2
Switch(config)#vlan 3 - создаем vlan 3
Switch(config-vlan)#name Server - название для vlan 3
Switch(config)#vlan 10 - создаем vlan 10
Switch(config-vlan)#name User's - название для vlan 10
Switch(config)#interface range fa0/1 - fa0/9 - настраиваем интерфейсы
                                         в сторону Пользователей
Switch(config-if-range)#description User's - описание интерфейса
Switch(config-if-range)#switchport mode access - настраиваем порт на
                                                тегированный режим
Switch(config-if-range)#switchport access vlan 10 - тегуем кадры
                                                10-й VLAN
Switch(config-if-range)#exit
Switch(config)#interface fa0/10 - настраиваем интерфейсы в сторону
                                Сервера
Switch(config-if)#description Server - описание интерфейса
Switch(config-if)#switchport mode access - настраиваем порт на
                                                тегированный режим
Switch(config-if)#switchport access vlan 3 - тегуем кадры 3 VLAN
Switch(config-if)#exit
Switch(config)#interface fa0/20 - настраиваем интерфейсы в сторону
                                Админа
Switch(config-if)#description Admin - описание интерфейса
Switch(config-if)#switchport mode access - настраиваем порт на
                                                тегированный режим
Switch(config-if)#switchport access vlan 2 - тегуем кадры 2 VLAN
Switch(config-if)#exit
Switch(config)#interface fa0/24 - настраиваем интерфейсы в сторону
                                маршрутизатора
Switch(config-if)#description Router - описание интерфейса
Switch(config-if)#switchport mode trunk - настраиваем порт на
                                                магистральный режим
Switch(config-if)#switchport trunk allowed vlan 2-3,10 – разрешаем
                                                кадры VLAN 2-3,10
Switch(config-if)#exit
Switch(config)#do write - сохраняем конфигурацию
```

Конфигурация для маршрутизатора:

```

Router>enable - переходим в расширенный режим
Router#configure terminal - переходим в режим конфигурации
Router(config)#interface fa0/0 - настраиваем порт в сторону коммутатора
Router(config-if)#description Switch - описание интерфейса
Router(config-if)#no shutdown - включаем интерфейс физически
Router(config-if)#exit
Router(config)#interface fa0/0.2 - настраиваем подинтерфейс для подсети
                               Админа
Router(config-subif)#description Admin - описание интерфейса
Router(config-subif)#encapsulation dot1q 2 - тегуем (помечаем) 2-ой VLAN
Router(config-subif)#ip address 172.16.0.1 255.255.255.0 - задаем шлюз
                               по умолчанию для Админа

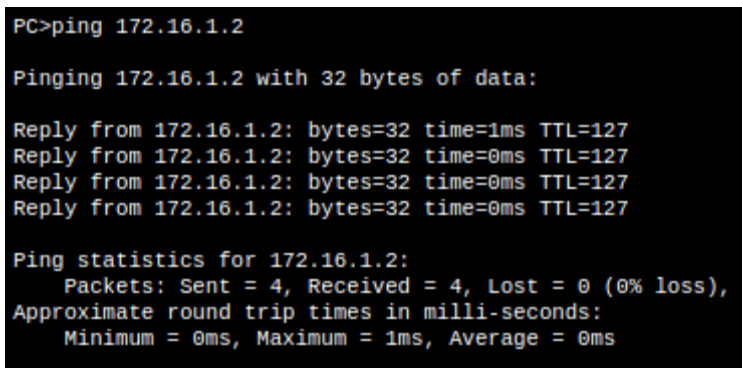
Router(config-subif)#exit
Router(config)#interface fa0/0.3 - настраиваем подинтерфейс для подсети
                               Серверов
Router(config-subif)#description Server - описание интерфейса
Router(config-subif)#encapsulation dot1q 3 - тегуем 3-ей VLAN
Router(config-subif)#ip address 172.16.1.1 255.255.255.0 - задаем шлюз
                               по умолчанию для Серверов

Router(config-subif)#exit
Router(config)#interface fa0/0.10 - настраиваем подинтерфейс для подсети
                               пользователей
Router(config-subif)#description User's - описание интерфейса
Router(config-subif)#encapsulation dot1q 10 - тегуем 10-ой VLAN
Router(config-subif)#ip address 172.16.2.1 255.255.255.0 - задаем шлюз по
                               умолчанию для Серверов

Router(config-subif)#exit

```

Запускаем пинг с пользовательского компьютера до сервера



```

PC>ping 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time=1ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127

Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

```

Как видно доступ есть. Нам же необходимо, чтобы доступ имел только админ. Для этого нам необходимо создать список доступа (пусть он будет

иметь порядковый номер 10), в котором мы разрешим всем пакетам от администратора (172.16.0.100) доступ в подсеть серверов (172.16.1.0/24). После чего применим это правило на подинтерфейсе fa0/0.3 (для серверов) для всех исходящих пакетов.

Router(config)#access-list 10 permit host 172.16.0.100 - создаем список доступа, в котором разрешаем хосту админа
 Router(config)#interface fa0/0.3 - настраиваем подинтерфейс для Серверов
 Router(config-subif)#ip access-group 10 out - применяем настройки списка доступа на подинтерфейсе

Тестируем настройки. Запускаем пинг с пользовательского компьютера в сторону сервера.

```
PC>ping 172.16.1.2
Pinging 172.16.1.2 with 32 bytes of data:
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Выдается сообщение Destination host unreachable – хост назначения недоступен. Запускаем пинг с компьютера администратора.

```
PC>ping 172.16.1.2
Pinging 172.16.1.2 with 32 bytes of data:
Reply from 172.16.1.2: bytes=32 time=1ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=5ms TTL=127
Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 5ms, Average = 1ms
```

Пинг проходит – значит ACL настроили правильно. Что происходит, когда мы пингуем сервер с ноутбука администратора? Пакет сначала поступает на подинтерфейс fa0/0.2 маршрутизатора. На данном интерфейсе не настроены списки доступа значит пакет проходит далее. Маршрутизатор анализирует свою таблицу маршрутизации и видит, что подсеть серверов находится на подинтерфейсе fa0/0.3. Перед отправкой пакета маршрутизатор обнаруживает, что к данному интерфейсу прикреплен ACL 10. В данном списке доступа всего одна запись – разрешить отправку пакетов только хосту 172.16.0.100 (ноутбук админа). Маршрутизатор анализирует IP-пакет и обнаруживает адрес отправителя 172.16.0.100 после чего отправляет пакет в подсеть серверов. IP-пакет с любым

отличным от 172.16.0.100 будет отбрасываться, так как в конце ACL 10 стоит неявный deny any – запретить все.

В связи с тем, что пользователям в заданной сети необходимо иметь доступ к файловому хранилищу и веб-сайту, требуется использовать расширенные списки доступа. Однако перед этим был полностью ограничен доступ клиентам к серверу. Для исправления ситуации необходимы расширенные списки доступа, которые могут проверять IP-адреса источника/отправителя, тип протокола, UDP/TCP-порты. В заданной ситуации необходимо будет проверять номера портов. Если пользователь обращается к серверу по разрешенному порту, то маршрутизатор пропускает такой пакет. Разрешенные порты: 80 (HTTP – доступ к веб-сайту), 21 (FTP – доступ к файловому хранилищу). Протоколы HTTP и FTP работают поверх TCP. Также для распознавания доменных имен на сервере необходимо включить (поднять) DNS, который работает на порту 53.

Размещать расширенный список доступа будем на подинтерфейсе fa0/0.3. Но на этом интерфейсе уже размещен список доступа. Следует помнить правило: нельзя разместить более одного списка доступа на интерфейс. По этой причине придется удалить созданный ранее список доступа. Правило, созданное для администратора, перенесем в новый расширенный список с именем *Server-out*.

Конфигурация для маршрутизатора:

```
Router(config)#no access-list 10 permit host 172.16.0.100
- удаляем предыдущий список доступа
Router(config)#interface fa0/0.3
- настраиваем сабинтерфейс для Серверов
Router(config-subif)#no ip access-group 10 out
- удаляем предыдущие настройки списка доступа
Router(config-subif)#exit
Router(config)#ip access-list extended Server-out
- создаем расширенный список доступа
Router(config-ext-nacl)#permit ip host 172.16.0.100 host 172.16.1.2 - даем админу
полный доступ к серверу
Router(config-ext-nacl)#permit tcp any host 172.16.1.2 eq 80
- разрешаем любому хосту доступ по HTTP к серверу
Router(config-ext-nacl)#permit tcp any host 172.16.1.2 eq 21
- разрешаем любому хосту доступ по FTP к серверу
Router(config-ext-nacl)#permit tcp any host 172.16.1.2 eq 53
- разрешаем любому хосту доступ по DNS к серверу
Router(config-ext-nacl)#exit
Router(config)#interface fa0/0.3
```

```
Router(config-if)#ip access-group Server-out out
```

С компьютера админа пинг до сервера есть:

```
PC>ping -t 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127

Ping statistics for 172.16.1.2:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

С компьютера пользователя ответа на пинг нет:

```
PC>ping -t 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.

Ping statistics for 172.16.1.2:
    Packets: Sent = 8, Received = 0, Lost = 8 (100% loss),
```

Проверим с компьютера пользователя проходят ли DNS-запросы до сервера. Для этого запустим утилиту *nslookup* – которая определяет IP-адрес до доменному имени.

```
PC>nslookup test.site

Server: [172.16.1.2]
Address: 172.16.1.2

Non-authoritative answer:
Name: test.site
Address: 172.16.1.2
```

DNS-запросы проходят без проблем. Проверим доступ к нашему условному Web-сайту через браузер:



Напоследок подключимся к FTP-серверу:

```
PC>ftp 172.16.1.2
Trying to connect...172.16.1.2
Connected to 172.16.1.2
220- Welcome to PT Ftp server
Username:user
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

Подключение прошло успешно!