Лабораторная работа №4.
Списки управления доступом ACL

Теоретическая часть.

Список управления доступом (access control list ACL) это последовательный список правил, которые используются для разрешения или запрета потока пакетов внутри сети на основании информации, приведенной внутри списка. Без списка доступа все пакеты внутри сети разрешаются без ограничений для всех частей сети. Список доступа может быть использован для контроля распространения и получения информации об изменении таблиц маршрутов и, главное, для обеспечения **безопасности**. Политика безопасности в частности включает защиту от внешних атак, ограничения доступа между отделами организации и распределение загрузки сети.

Список доступа позволяет использовать маршрутизатор как межсетевой экран, брандмауэр, для запрета или ограничения доступа к внутренней сети из внешней сети, например, Интернет. Брандмауэр, как правило, помещается в точках соединения между двумя сетями.

Стандартный ACL

При использовании стандартных ACL, единственным критерием для определения того, что пакет разрешен или запрещён, является IP адрес источника этого пакета. Формат элемента списка доступа следующий

Router(config)#**access-list № permit | deny source-address source-mask,**

где № – целое число – номер списка доступа, source-address обозначает адрес источника пакета, source-mask – маска в инверсной форме, накладываемая на адрес, permit – разрешить прохождение пакета, deny – запретить прохождение пакета.

Число № определяет принадлежность элемента списка доступа к определённому списку доступа с номером №. Первая команда access-list определяет первый элемент списка доступа, вторая команда определяет второй элемент списка доступа и т.д. Маршрутизатор обрабатывает каждый определённый в нём список доступа по элементам сверху вниз. То есть, если адрес source-address пакета с учётом маски удовлетворяет условию элемента списка, то дальнейшие элементы списка маршрутизатор не обрабатывает. Следовательно, для избежания лишней обработки, элементы, определяющие более общие условия, следует помещать в начале списка. Внутри маршрутизатора может быть определено несколько списков доступа. Номер стандартного списка должен лежать в диапазоне 1 – 99. Маска в списке

доступа задаётся в инверсной форме, например маска 255.255.0.0 выглядит как 0.0.255.255.

Маршрутизаторы Cisco предполагают, что все адреса, не упомянутые в списке доступа в явном виде, запрещены. То есть в конце списка доступа присутствует невидимый элемент

```
Router(config)#access-list # deny 0.0.0.0 255.255.255.255
```

Так, если мы хотим разрешить только трафик от адреса 1.1.1.1 и запретить весь остальной трафик достаточно в список доступа поместить один элемент

```
Router(config)#access-list 77 permit 1.1.1.1 0.0.0.0.
```

Здесь предполагается, что мы организовали список доступа с номером 77.

Рассмотрим возможность применения стандартных списков доступа для диапазона адресов. Возьмём к примеру диапазон 10.3.16.0 – 10.3.31.255. Для получения инверсной маски можно вычесть из старшего адреса младший и получить 0.0.15.255. Тогда пример элемента списка можно задать командой

```
Router(config)#access-list 100 permit 10.3.16.0 0.0.15.255
```

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

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

или

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

Список доступа может быть применён либо как входной (in) либо как выходной (out). Когда вы применяете список доступа как входной, то маршрутизатор получает входной пакет и сверяет его входной 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 вы применяете список доступа как выходной, то роутер получает входной пакет, маршрутизирует его на интерфейс назначения и только тогда обрабатывает входной адрес пакета согласно элементам списка доступа этого интерфейса. Далее маршрутизатор либо разрешает пакету покинуть интерфейс либо отбрасывает его согласно разрешающим и запрещающим элементам списка соответственно. Так, созданный ранее список с номером 77 применяется к интерфейсу Ethernet 0 маршрутизатора как входной список 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
```

Отменяется список на интерфейсе с помощью команды **no**

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

Приступим к созданию более сложных списков доступа. Рассмотрим сеть на рисунке 1.

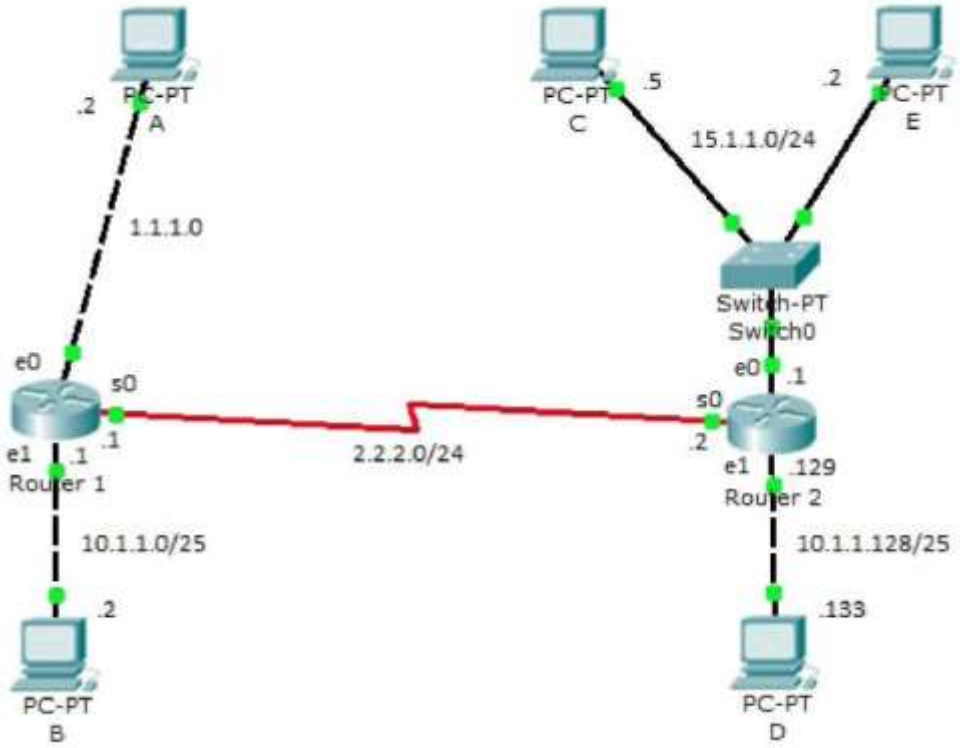

Рисунок 1.

Разрешим все пакеты, исходящие из сети 10.1.1.0 /25 (10.1.1.0 255.255.255.128) , но запретим все пакеты, исходящие из сети 10.1.1.128 /25 (10.1.1.128 255.255.255.128). Мы также хотим 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 мы разрешаем. Списку дадим 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 отсутсвие 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 мы выполнили поставленную задачу.

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. Важно отметить вид инверсной маски 0.0.0.127 для этой сети. Эта маска указывает, что мы не должны брать в рассмотрение последние семь бит четвертого октета адреса, которые назначены для адресации в данной подсети. Маска для этой сети 255.255.255.128, которая говорит, что последние семь бит четвертого октета определяют адресацию 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. Мы вначале требуем запретить все пакеты, исходящие из сети 15.1.1.0 255.255.255.0 и лишь затем разрешить пакеты с адресом 15.1.1.5. Если в командах, определяющих список доступа мы, переставим 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 запрещён 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, имеем следующую последовательность действий для воплощения списка доступа.

1. Определить критерии и ограничения для доступа.

2. Воплотить их с помощью команд access-list, создав список доступа с определённым номером.

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

Остановимся на последнем пункте. В общем случае стандартный список доступа следует помещать как можно ближе к точке назначения, а не к источнику пакетов. Хотя могут быть исключения. Так как стандартный список доступа работает только с исходными адресами, то не всегда возможна детальная конфигурация. Требуется приложить усилия, чтобы избежать возникновения не желаемых конфигураций доступа. Если список помещён вблизи источника пакетов, то очень вероятно, что доступ к устройствам, на которых не осуществляется никакая конфигурация доступа, будет затруднён.

Конкретизируем политику безопасности для сети на рисунке 1. Наша цель создать политику для компьютера A (адрес 1.1.1.2 сеть 1.1.1.0/24), которая изо всех устройств локальной сети 15.1.1.0 /24 в которую входит компьютер C (15.1.1.5) разрешит доступ к компьютеру A лишь самого компьютера C. Мы также хотим создать политику, запрещающую удалённый доступ к компьютеру A из любого устройства локальной сети 10.1.1.128 / 25 компьютера D (10.1.1.133). Весь остальной трафик мы разрешаем. На рисунке 1 компьютер PC5 (15.1.1.5) играет роль произвольного отличного от компьютера C представителя локальной сети 15.1.1.0/24.

Размещение списка критично для воплощения такой политики. Возьмём созданный ранее список с номером 2. Если список сделать выходным на последовательном интерфейсе маршрутизатора 2, то задача для компьютера A 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 Ethernet A интерфейс маршрутизатора 1, то задача будет выполнена безо всяких побочных эффектов.

Расширенные ACL

Со стандартным ACL вы можете указывать только адрес источника, а маска не обязательна. В расширенных ACL вы должны указать и адрес приёмника и адрес источника с масками. Можете добавить дополнительную протокольную информацию для источника и назначения. Например, для TCP и UDP разрешено указывать номер порта, а для ICMP разрешено указывать тип сообщения. Как и для стандартных ACL, можно с помощью опции log осуществлять лог.

Общая форма команды для формирования строки списка расширенного доступа

access-list access-list-number {permit | deny} protocol source source-wildcard [operator source-port] destination destination-wildcard [operator

destination-port] [precedence precedence-number] [tos tos] [established] [log | log-input],

где access-list-number -100-199|2000-2699, protocol - ip, icmp, tcp, gre, udp, igrp, eigrp, igmp, ipinip, nos и ospf. Для порта source-port или destination-port можно использовать номер порта или его обозначение bgp, chargen, daytime, discard, domain, echo, finger, ftp, ftp-data, gopher, hostname, irc, klogin, kshell, lpd, nntp, pop2, pop3, smtp, sunrpc, syslog, tacacs-ds, talk, telnet, time, uucp, whois и www. Operator это eq (равно), neq (не равно), gt (больше чем), lt (меньше чем), range (указывается два порта для определения диапазона).

Как и для стандартных ACL расширенный ACL следует привязать к интерфейсу либо для входящего на интерфейс трафика

Router(config-if)# ip access-group №ACL in

либо для выходящего из интерфейса трафика

Router(config-if)# ip access-group №ACL out

здесь №ACL - номер списка. Примеры элементов расширенного ACL Разрешить SMTP отовсюду на хост

Router(config)# access-list 111 permit tcp any host 172.17.11.19 eq 25

Разрешить telnet отовсюду на хост

Router(config)# access-list 111 permit tcp any host 172.17.11.19 eq 23

Расширенный ACL позволяет очень тонко настроить права доступа.

Именованные ACL

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

Router(config)# ip access-list extended ACL_name

и далее команды для создания элементов списка именно этого неименованного 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

ACL обрабатываются сверху вниз. Наиболее часто повторяющийся трафик должен быть обработан в начале списка. Как только обрабатываемый списком 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.

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

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

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

Для редактирования числовых ACLs следует его уничтожить и создать заново или изменить список офлайн и загрузить в устройство с помощью.

Практическая часть.

1. Создадим и загрузим в симулятор топологию, изображённую на рисунке 2.

Рисунок 2.

Назначим адреса интерфейсам (маска 255.255.255.240) согласно таблице. Не забудьте на DCE устройстве последовательного соединения задать синхронизацию.

	Router 2	Router 1	Router 4
Ethern	24.17.2.2	24.17.2.1	
Serial		24.17.2.17	24.17.2.18

Осуществим конфигурацию RIP маршрутизации

Для Router1

```
Router1(config)#router rip
Router1(config- router)#version 2
Router1(config- router)#network 24.0.0.0
```

На Router2

```
Router2(config)#router rip
Router1(config- router)#version 2
Router2(config- router)#network 24.0.0.0
```

и на Router4

```
Router4(config)#router rip
Router1(config- router)#version 2
Router4(config- router)#network 24.0.0.0
```

Проверьте свою сеть с помощью команды ping и, в частности, что вы можете пинговать интерфейс Ethernet0 (24.17.2.2) маршрутизатора 2 из маршрутизатора 4

```
Router4#ping 24.17.2.2
```

Создадим стандартный список доступа, который не позволит пинговать маршрутизатор 2 из маршрутизатора 4. Для этого блокируем единственный адресс 24.17.2.18 маршрутизатора 4 и разрешим остальной трафик. Список создадим на маршрутизаторе 2 командами

```
Router2(config)#access-list 1 deny 24.17.2.18 0.0.0.0
Router2(config)#access-list 1 permit 0.0.0.0 255.255.255.255
```

Применим список к интерфейсу Ethernet маршрутизаторе 2

```
Router2(config)#interface FastEthernet0/0
Router2(config-if)#ip access-group 1 in
```

Проверим, что список доступа запущен. Для этого просмотрим работающую конфигурацию

```
Router2#show running-config
```

Мы также можем видеть, что список применён к интерфейсу, используя команду “show ip interface”. Найдите в выводимой информации с строку “Innbound access list is 1”.

```
Router2#show ip interface
```

Команда “show access-lists” покажет нам содержимое созданного списка доступа.

```
Router2#show access-lists
Standard IP access list 1
    deny host 24.17.2.18
    permit any (4 match(es))
```

Отметим, что host 24.17.2.18 равносильно 24.17.2.18 0.0.0.0. Теперь при попытке пинговать интерфейс Ethernet0 (24.17.2.2) роутера 2 из роутера 4

```
Router4#ping 24.17.2.2
Type escape sequence to abort.
Sending 5, 100-liyte ICMP Echos to 24.17.2.2, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
```

Получим строку “UUUUU”, которая означает, что список доступа работает корректно.

2. Создадим и загрузим в симулятор топологию на рисунке 3,

Рисунок 3.

Назначим адреса интерфейсам (маска 255.255.255.0) согласно таблице

	Router 2	Router 1	Router 3	Router 4
Ethernet 0	160.10.1.2	160.10.1.1	175.10.1.2	180.10.1.2
Ethernet 1		175.10.1.1	180.10.1.1	

Осуществим конфигурацию OSPF маршрутизации Для Router1

```
Router1(config)#router ospf 1
Router1(config- router)#network 160.10.1.0 0.0.0.255 area 0
Router1(config- router)#network 175.10.1.0 0.0.0.255 area 0
```

Для Router2

```
Router2(config)#router ospf 1
Router2(config- router)#network 160.10.1.0 0.0.0.255 area 0
end
```

Для Router3

```
Router3(config) #router ospf 1
Router3(config- router)#network 175.10.1.0 0.0.0.255 area 0
Router3(config- router)#network 180.10.1.0 0.0.0.255 area 0
```

Для Router4

```
Router4(config )#router ospf 1
Router4(config- router)#network 180.10.1.0 0.0.0.255 area 0
```

Для проверки пропингуйте крайние точки

```
router2#ping 180.10.1.2
```

и

```
router4#ping 160.10.1.2
```

Создадим стандартный список доступа для фильтрации трафика, приходящего на интерфейс ethernet0 1-го маршрутизатора router1 и разрешает трафик от подсети 175.10.1.0 (router3) и блокирует трафик от других устройств.

```
router1(config)#access-list 1 permit 175.10.1.0 0.0.0.255
```

Проверьте, что он создался

router1#**show access-list**
```
Standard IP access list 1
permit 175.10.1.0 0.0.0.255
```

Присоедините список как входной к интерфейсу Ethernet 1

router1(config)#**interface FastEthernet1/0** router1(config-if)#**ip access-group 1 in**

Проверьте присоединение командой

router1# **show running-config**

Проверьте связь между 3 и 2 маршрутизаторами и между 4 и 2 .

router3# **ping 160.10.1.2** router4# **ping 160.10.1.2**

Связь между 3 и 2-м роутерами должна быть, а между 4 и 2 - нет. Изменим список доступа и разрешим трафик от подсети 180.10.1.0 (router4) и блокируем трафик от других устройств.

router1(config)# **no access-list 2**
router1(config)# **access-list 2 permit 180.10.1.0 0.0.0.255**

Проверьте, что он изменился

router1#**show access-list**
```
Standard IF access list 1
perir.it 130.10.1.0 0.0.0.2 55
```

Присоедините список как входной к интерфейсу Ethernet 1

router1(config)#**interface FastEthernet1/0**
router1(config-if)#**ip access-group 1 in**

Проверьте присоединение командой

router1# **show running-config**

Проверьте связь между 3 и 2 маршрутизаторами и между 4 и 2.

router3# **ping 160.10.1.2**
router4# **ping 160.10.1.2**

Связь между 4 и 2-м роутерами должна быть, а между 3 и 2 - нет.

3. Осуществите и проверьте конфигурацию IP для сети на рисунке 1 и примените OSPF для организации динамической маршрутизации.

Для маршрутизатора router 1

```
router1(config)#router ospf 1
router1(config-router)#network 2.2.2.0 0.0.0.255 area 0
router1(config-router)#network 1.1.1.0 0.0.0.255 area 0
router1(config-router)#network 10.1.1.0 0.0.0.127 area 0
```

Для маршрутизатора router 2

```
Router2(config)#router ospf 1
Router2(config-router)#network 10.1.1.128 0.0.0.127 area 0
Router2(config-router)#network 15.1.1.0 0.0.0.255 area 0
Router2(config-router)#network 2.2.2.0 0.0.0.255 area 0
```

Проверте работоспособность сети: вы должны из любого устройства пинговать любой интерфейс. Или проще: все компьютеры A, B, C, D, PC5 должны взаимно попарно пинговаться.

Создадим список доступа из теоретической части

3.1 На маршрутизаторе router 1 создадим список доступа

```
router1(config)#access-list 2 deny   10.1.1.128 0.0.0.127
router1(config)#access-list 2 permit host 15.1.1.5
router1(config)#access-list 2 deny   15.1.1.0 0.0.0.255
router1(config)#access-list 2 permit 0.0.0.0 255.255.255.255
```

и применим его к интерфейсу Ethernet0 как выходной

```
router1(config)#interface FastEthernet0/0
router1(config-if)#ip access-group 2 out
```

Создать скриншот результата выполнения команды

```
router1#show access-list
```

Попарно пропингуем A, B, C, PC5, D. В результате должна получиться следующая матрица доступа

Таблица 1

Из\В	A	B	C	E	D
A	+	+	+	-	-
B	+	+	+	+	+
C	+	+	+	+	+
E	-	+	+	+	+
D	-	+	+	+	+

Видим, что политика безопасности из теоретической части полностью реализована.

3.2 Удалим ACL с интерфейса e0 и применим как входной к интерфейсу s0

```
router1(config)#interface fa0/0
router1(config-if)#no ip access-group 2 out
router1(config-if)#int s2/0
router1(config-if)#ip access-group 2 in
```

Попарно пропингуем A, B, C, PC5, D. В результате должна получиться следующая матрица доступа

Таблица 2

Из\В	A	B	C	E	D
A	+	+	+	-	-
B	+	+	+	-	-
C	+	+	+	+	+
E	-	-	+	+	+
D	-	-	+	+	+

Видим, что теперь трафик между сетями 10.1.1.0/25 и 10.1.1.128/25 запрещен. Невозможен также трафик между сетью 10.1.1.0/25 и сетью 15.1.1.0/24 за исключением компьютера C с адресом 15.1.1.5.

4. Используем топологию и конфигурацию пункта 1 этой лабораторной работы Отменим конфигурацию доступа, сделанную в пункте 1

```
Router2(config)#no access-list 1 deny 24.17.2.18 0.0.0.0
Router2(config)#no access-list 1 permit 0.0.0.0 255.255.255.255
```

Применим список к интерфейсу Ethernet маршрутизаторе 2

```
Router2(config)#interface fa0/0
Router2(config-if)# no ip access-group 1 in
```

Разрешим заходить на router1 телнетом на его два интерфейса с паролем router1

```
Router1(config)#line vty 0 4
Router1(config-line)#login
Router1(config-line)#password router1
```

Наши EACL будут делать пару различных вещей. Первое мы разрешим только telnet из подсети последовательного соединения 24.17.2.16/240 для входа на router1

```
router1(conf)#access-list 101 permit tcp 24.17.2.16 0.0.0.15 any eq telnet log
```

Опция log заставит маршрутизатор показывать выход при срабатывании списка доступа.

Разрешим на маршрутизаторе router1 весь трафик из Ethernet 0 подсети 24.17.2.0/240

```
router1(conf)#access-list 102 permit ip 24.17.2.0 0.0.0.15 any
```

Проверим установку списков

```
router1#show access-list
Extended IP access list 101
    permit tcp 24.17.2.16 0.0.0.15 any eq telnet log (0 matches)
Extended IP access list 102
    permit ip 24.17.2.0 0.0.0.15 any log (1 matches)
```

Теперь применим списки к интерфейсам для входящих пакетов

```
router1(conf)# interface Serial2/0
router1(conf-if)# ip access-group 101 in
router1(conf-if)# interface fa0/0
router1(conf-if)# ip access-group 102 in
```

Для проверки, что EACL присутствуют на интерфейсах, используйте команду

```
router1#show running-config
```

или

```
router1#show ip interface
```

Проверим функционирования EACL. Присоединимся к router4 и попытаемся безуспешно пропинговать интерфейс Serial2/0 на router1

```
router4#ping 24.17.2.17
```

EACL номер 101 блокировал ping. Но должен разрешить telnet

```
router4#telnet 24.17.2.17
```

Успешно. Введём пароль router1. Промпт router4# изменился на router1>. Нажав одновременно ctrl-shift-6 и затем 6, вернёмся на router4. О срабатывании EACL 101 на router1 нам укажет лог

```
00:06:50:  %SEC-6-IPACCESSL0GDP: list 101 permitted TCP 24.17.2.1B -> 24.17.2.17 (B/O), 5 packets
```

Посмотрим номер сессии и убьём телнет соединение

router4#**show sess**
router4# **disconnect 1**

Присоединимся к router2 и посмотрим, можем ли мы пропинговать интерфейс Serial0 на router4.

Router2# **ping 24.17.2.18**

Почему неудачно? Пакет стартует в Router2, идёт через Router1 (о срабатывании EACL 102 на router1 нам укажет лог

```
00:03:2Э:  %SEC-6-IPACCESSL0GDP: list 102 permitted IP 24.17.2.2 -> 24.17.2.18 (8/0), 5 packets
```

и приходит на Router4. На Router4 он переформируется и отсылается обратно к Router1. Когда Router4 переформирует пакет, адрес источника становится адресом приёмника и адрес приёмника становится адресом источника. Когда пакет приходит на интерфейс Serial0 на router1 он отвергается, так как его адрес источника равен IP адресу интерфейса Serial0 на router4 24.17.2.17, а здесь разрешён лишь tcp.

Присоединимся к router2 и посмотрим, можем ли мы пропинговать интерфейс Ethernet0 на router1.

router2#**ping 24.17.2.1**

Успешно. Аналогично и для телнета

router2#**telnet 24.17.2.1**

EACL работают успешно. О срабатывании EACL 102 на router1 нам укажет лог.

```
00:05:22:  %SEC-6-IPACCESSL0GDP: list 102 permitted IP 24.17.2.2 -> 24.17.2.1 (8/0), 5 packets
```

Заметим, что лог так же постоянно показывает RIP обновления

```
00:06:42:  %SEC-6-IPACCESSL0GDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
00:06:12:  %SEC-6-IPACCESSL0GDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
00:07:42:  %SEC-6-IPACCESSL0GDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
00:07:12:  %SEC-6-IPACCESSL0GDP: list 102 permitted IP 24.17.2.2 -> 255.255.255.255 (8/0), 5 packets
```

5. Именованные ACL

Отменим на router1 привязку EACL к интерфейсам

```
router1(conf)# interface Serial0
router1(conf-if)# no ip access-group 101 in
router1(conf-if)# interface Ethernet0
router1(conf-if)#no ip access-group 102 in
```

и отменим на router1 EACL

```
router1(conf)#no access-list 101
router1(conf)#no access-list 102
```

Поставим задачу запретить по всей сети лишь пинги от router4 на router2. Список доступа можно расположить и на router1 и на router2. Хотя рекомендуют располагать ACL ближе к источнику (для сокращения трафика), в этом примере расположим именованный список с именем deny_ping на router2.

```
router2(config)#ip access-list extended deny_ping
router2(config-ext-nacl)#deny icmp 24.17.2.18 0.0.0.0 24.17.2.2 0.0.0.0 log
router2(config- ext-nacl)# permit ip any any log
```

Первая команда указывает, что мы создаём именованный расширенный список доступа с именем deny_ping. Вторая команда указывает на запрещение ICMP трафика с адресом источника строго 24.17.2.18 и адресом приёмника строго 24.17.2.2. Третья команда разрешает остальной IP трафик.

Проверим создание списка

```
router2#show access-list
Extended IP access list deny_ping
    deny   icmp host 24.17.2.18 host 24.17.2.2 log (0 matches)
    permit ip any any log (0 matches)
```

Всё правильно, мы видим в первой строке просто другую форму представления команды deny icmp 24.17.2.18 0.0.0.0 24.17.2.2 0.0.0.0 log.

Применим список для входного трафика интерфейса Ethernet0 на router2

```
Router2(conf)#interface Ethernet0
Router2(conf-if)#ip access-group deny_ping in
```

Присоединимся к router4 и пропингуем роутер2

```
router4#ping 24.17.2.2
```

Неудача. Присоединимся к router1 и пропингуем роутер2

```
Router1#ping 24.17.2.2
```

Успех. Присоединимся к router2 и посмотрим на два отдельных лог-сообщения: первое о запрещении пинга от router4 и второе о разрешении пинга от router1

```
100:11:18: &SEC-6-IPACCESSL0GDP: list 0 permitted IP 24.17.2.1 -> 24.17.2.2 (8/0), 5 packets
100:12:30:  &SEC-6-IPACCESSL0GDP: list 0 permitted IP 24.17.2.1 -> 255.255.255.255 (8/0), 5
packets
```

6. Рассмотрим более сложные вопросы расширенных списков доступа. Создадим топологию

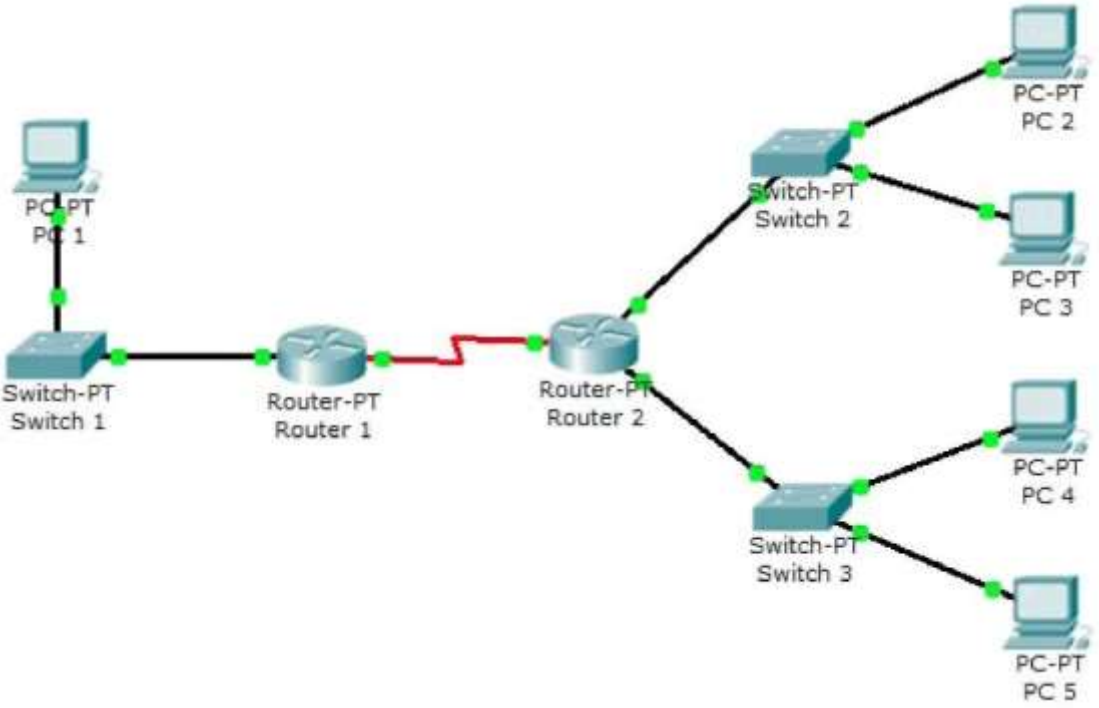

Рисунок 4.

Используйте коммутаторы модели 1912. Маршрутизатор Router1 - модели 805. Маршрутизатор Router2 - модели 1605. Назначим IP адреса маршрутизаторам

	Router1	Router2
Fa0/0	1.1.3.1/24	1.1.1.129/25
Fa1/0		1.1.1.1/25
Serial2/0	1.1.2.1/24	1.1.2.2/24

и компьютерам

Hostname	IP на ethernet0	Шлюз
PC1	1.1.3.2 255.255.255.0	1.1.3.1
PC2	1.1.1.130 255.255.255.128	1.1.1.129
PC3	1.1.1.131 255.255.255.128	1.1.1.129
PC4	1.1.1.2 255.255.255.128	1.1.1.1
PC5	1.1.1.3 255.255.255.128	1.1.1.1

На Router1 и Router2 конфигурируем RIP

Router(config)#**router rip** Router(config-router)#**network 1.0.0.0**

Интерфейсы всех устойств должны пинговаться со всех устройств.

6.1. Список доступа сеть-сеть.

Создадим список, который разрешает трафик от локальной сети компьютеров PC4 и PC5 в локальную сеть компьютера PC1 и запрещает трафик от локальной сети компьютеров PC2 и PC3 в локальную сеть компьютера PC1. Так как трафик приходит от router2 к router1 , то следует поместить список доступа на интерфейс serial2/0 router1 для входного трафика

Router1(conf)#**access-list 100 permit ip 1.1.1.0 0.0.0.127 1.1.3.0 0.0.0.255 log**

Router1(conf)#**access-list 100 permit ip 1.1.2.0 0.0.0.255 any log**

Первая команда непосредственно решает поставленную задачу, а вторая разрешает широковещание RIP протоколов. Проверим создание

Router1#**show access-list**

```
Extended IP access list 100
    permit ip 1.1.1.0 0.0.0.127 1.1.3.0 0.0.0.255 log (0 matches)
    permit ip 1.1.2.0 0.0.0.255 any log (11 matches)
```

Применим список доступа к интерфейсу.

Router1(conf)#interface Serial2/0

Router1(conf-if)#ip access-group 100 in

Для тестирования списка доступа, попытайтесь пропинговать PC1 от PC2, PC3, PC4 и PC5.

PC#**Ping 1.1.3.2**

Для PC2 и PC3 пинги не пойдут. Для PC4 и PC5 пинги пойдут. Список доступа работает. Посмотрите логи на router1

```
101:31:39:   %SEC-6-IPACCESSL0GDP: list 100 permitted IP 1.1.1.2 -> 1.1.3.2 (8/0), 5
packets
```

6.2. Список доступа хост-хост.

Создадим на router2 список доступа, который блокирует доступ к PC5 только с PC2. Контролировать попытки доступа можно по логам на router2.

```
Router2(conf)# access-list 101 deny ip 1.1.1.130 0.0.0.0 1.1.1.3 0.0.0.0 log
Router2(conf)# access-list 101 permit ip any any
```

Проверим создание

```
Router2#show access-list
Extended IP access list 101
    deny ip host 1.1.1.130 host 1.1.1.3
    permit ip any any
```

Применим список доступа к fast Ethernet интерфейсу router2

```
Router2(conf)#interface FastEthernet0/0
Router2(conf-if)#ip access-group 101 in
```

Присоединитесь к PC2 и проверьте, что вы не можете пиновать PC5

```
PC2# Ping 1.1.1.3
```

На router2 появится лог

```
01:51:44:  %SEC-6-IPACCESSL0GDP: list 101 denied IP 1.1.1.130 -> 1.1.1.3 (8/0), 5 packets
```

Присоединитесь к PC3 и проверьте, что вы можете пиновать PC5.

```
PC3# Ping 1.1.1.3
```

На router2 появится лог

```
01:54:41:  %SEC-6-IPACCESSL0GDP: list 101 permitted IP 1.1.1.131 -> 1.1.1.3 (8/0), 5 packets
```

6.3. Список доступа сеть-хост.

Вначале удалим предыдущие списки доступа с интерфейсов Router1 и Router2.

```
Router1(conf)#interface Serial2/0
Router1(conf-if)#no ip access-group 100 in
```

и

```
Router2(conf)#interface FastEthernet0/0
Router2(conf-if)#no ip access-group 101 in
```

Создадим расширенный список доступа, который блокирует весь трафик к PC1 из локальной сети компьютеров PC2 и PC3. Так как мы блокируем весь трафик, то будем использовать IP протокол.

```
Router2(conf)#access-list 102 deny ip 1.1.1.128 0.0.0.127 1.1.3.2 0.0.0.0 log
Router2(conf)#access-list 102 permit ip any any
```

Проверим создание

```
Router2#show access-list
Extended IP access list 102
    deny ip 1.1.1.128 0.0.0.127 host 1.1.3.2
    permit ip any any
```

Применим список к исходящему трафику на интерфейсе Serial2/0 Router2

```
Router2(conf)#interface Serial2/0
Router2(conf-if)#ip access-group 102 out
```

Для проверки списка попытайтесь пропинговать PC1 (1.1.3.2) из PC2 и PC3. Пинги не пройдут. Симулятор почему-то не даёт лог на консоли Router2. Но эффект выможете увидеть так

```
Router2#sh ac
Extended IP access list 102
    deny ip 1.1.1.128 0.0.0.127 host 1.1.3.2 (8 match(es))
    permit ip any any

Router2#sh ac
Extended IP access list 102
    deny ip 1.1.1.128 0.0.0.127 host 1.1.3.2 (16 match(es))
    permit ip any any
```

Вы видите после каждого неудачного пинга количество отслеженных (matches) пакетов возрастает.

Контрольные вопросы

1. Что такое ACL?
2. Какой адрес является критерием для разрешения/запрещения пакета?
3. Где применяются ACL?
4. Как задать элемент ACL и что такое инверсная маска?
7. Что фильтруют расширенные ACL?
8. Какую дополнительную функциональность имеют расширенные ACL по сравнению со стандартными?
9. Можно ли, используя расширенные ACL, наложить ограничения на трафик к определённой TCP/IP службе?
10. Опишите процедуру создания именованного ACL.

Ход работы

1. Изучить теоретическую и практическую часть.

2. Сдать теорию работы путём предоставления ответа на контрольные вопросы.

3. Выполнить три пункта практической части. Проверить доступ согласно таблице1, а затем таблице 2.

4. Показать, что сеть удовлетворяет матрице доступа из таблицы 1.

5. Показать, что сеть удовлетворяет матрице доступа из таблицы 2.

6. Выполните в Packet Tracer задание для получения повышенного бала.

7. Построить матрицу доступа согласно варианту и показать, что сеть удовлетворяет этой матрице.

8. Оформите отчёт.

Содержание отчёта.

1. Скриншоты топологий, созданных в практической части.

2. Все скриншоты, созданные в практической части.

3. Конфигурации всех маршрутизаторов созданных в практической части.