Министерство образования и науки РФ

Севастопольский государственный университет

ИССЛЕДОВАНИЕ СПОСОБОВ НАЗНАЧЕНИЯ СПИСКОВ КОНТРОЛЯ ДОСТУПА В ЛОКАЛЬНЫХ КОМПЬЮТЕРНЫХ СЕТЯХ

Методические указания
к выполнению лабораторной работы №3
по дисциплине

“Архитектура

инфокоммуникационных систем и сетей”

Для студентов, обучающихся по направлению 09.03.02
"Информационные системы и технологии"
по учебному плану подготовки бакалавров
дневной и заочной форм обучения

Севастополь
2016

Цель работы: исследовать способы назначения стандартных и расширенных списков контроля доступа (ACL).

1. Списки доступа

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.

1.1. 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 избегания 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 либо 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
```

Рассмотрим принцип создания более сложных списков доступа. Пусть имеем сеть, представленную на рис. 3.1. 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 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 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.

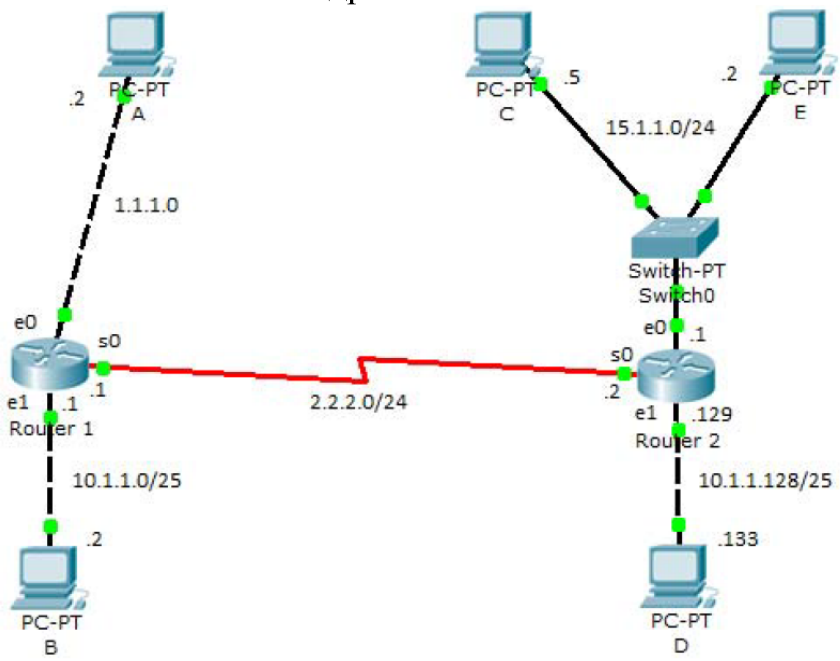


Рис. 3.1. Пример топологи сети для создания сложных списков доступа

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 рис. 3.1. 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 рис. 3.1 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.

1.2. 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
sourcewildcard [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 (общепринятые номера портов представлена в табл. 3.1).

`precedence precedence` - `(0..7)` Первые 3-и бита поля TOS (тоже самое можно сделать через TOS).

`tos tos` - `(0..15)` Поле TOS IPv4 пакета (Type of service)

`log` – логирование на консоль (какой ACL, протокол, откуда+куда пакет пришел)

`log-input` – тоже что и log + интерфейс + MAC адрес отправителя.

Таблица 3.1

Основные общепринятые номера портов

Порт / Протокол	Описание
20/TCP	протокол FTP – данные
21/TCP	протокол FTP – команды
22/TCP,UDP	протокол SSH
23/TCP,UDP	протокол Telnet
25/TCP,UDP	протокол SMTP
20/TCP	протокол FTP – данные
21/TCP	протокол FTP – команды
35/TCP,UDP	протокол приватного сервера печати printer server
53/TCP,UDP	Domain Name System (DNS)
80/TCP	Hypertext Transfer Protocol (HTTP)
109/TCP	Post Office Protocol 2 (POP2)
110/TCP	Post Office Protocol 3 (POP3)
156/TCP,UDP	SQL Service
161/TCP,UDP	Simple Network Management Protocol (SNMP)
443/TCP	HTTP поверх TLS/SSL (HTTPS)
445/TCP	Microsoft-DS Active Directory, Windows shares
520/UDP	Routing – RIP
1433/TCP,UDP	Microsoft SQL Server – Server
1434/TCP,UDP	Microsoft SQL Server – Monitor

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.

1.2.1. 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 телнет отовсюду на хост

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

`Any` – это специальное слово, которое означает адрес сети и обратную маску `0.0.0.0 0.0.0.0` и означает, что под правило подпадают абсолютно все узлы из любых сетей. Другое специальное слово – **`host`** – оно означает маску `255.255.255.255` – то есть именно один единственный указанный адрес.

Расширенный ACL позволяет очень тонко настроить права доступа.

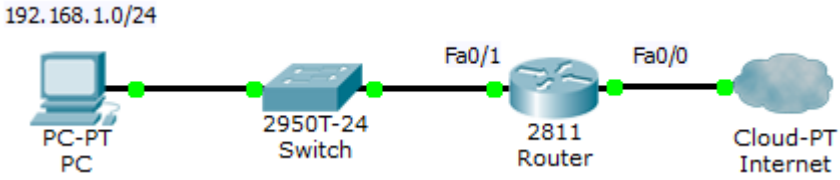
1.2.2. Применение established в ACL

Ключевое слово **`established`** используется в расширенных ACL для определения, принадлежит ли трафик к открытой TCP сессии. Маршрутизатор проверяет, соответствующий бит в заголовке TCP и принимает решение относительно того, относится ли трафик к уже установленному соединению.

Типичное использование established – организация доступа к интернету для сотрудников, чтобы извне нельзя было обращаться ко внутренней сети, но при этом ответы от веб серверов проходили вовнутрь нормально.

Established имеет ряд недостатков. Основной из них – работа только с протоколом TCP, так как используется его внутренний флаг. Если требуется работа с другими протоколами, следует использовать для этих же целей зеркальные (reflexive) ACL.

Допустим, есть топология:



Внутренняя сеть 192.168.1.0/24, надо обеспечить доступ из неё в интернет так чтобы ответы от серверов из интернета работали, но при этом обратиться вовнутрь извне было нельзя. Внутренняя сеть подключена к Fa0/1, внешняя – Fa0/0.

Настройка будет выглядеть следующим образом:

```
Router(config)#access-list 101 permit tcp 192.168.1.0 0.0.0.255 any eq 80
Router(config)#access-list 102 permit tcp any eq 80 192.168.1.0 0.0.0.255 established
Router(config)#interface fa0/1
Router(config-if)#ip access-group 101 in
Router(config-if)#interface fa0/0
Router(config-if)#ip access-group 102 in
```

Расширенный ACL 101 служит для выпуска трафика из пользовательской сети, он настроен на вход на интерфейса fa0/1. Он уничтожает весь трафик кроме того, что идёт из сети на любой адрес на 80-ый порт.

ACL 102 используется на Fa0/0 – на вход. Когда из интернета приходит пакет, он проверяется сразу же этим ACL. Пропускается только трафик, идущий с 80-го порта на удалённом сервере (ответы от веб-серверов), только во внутреннюю сеть, и, самое главное только established трафик, то есть только трафик в рамках сессии которую установили мы изнутри.

В итоге, изнутри можно обращаться к сайтам и получать от них ответы, но если злоумышленник захочет подключиться к компьютеру внутри нашей сети (будучи сам снаружи), у него это не получится, даже если он будет пытаться подключиться с 80-го порта, так как при подключении он будет устанавливать новое соединение, и установить он его не сможет, так как в первых TCP сегментах не будет стоять необходимый флаг, соответственно, не произойдёт TCP-рукопожатие.

1.2.3. Reflexive ACL – зеркальные списки контроля доступа

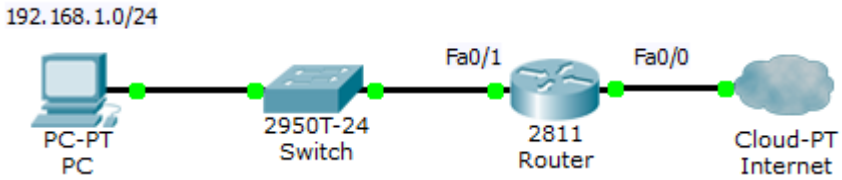
Reflexive ACL – зеркальные списки контроля доступа, позволяют запоминать, кто обращался из нашей сети наружу (с каких адресов, с каких портов, на какие адреса, на какие порты) и автоматически формировать зеркальный ACL, который будет пропускать обратный трафик извне вовнутрь только в том случае, если изнутри было обращение к данному ресурсу.

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.

Технология эта напоминает внешне использование ключевого слова established, но имеется ряд отличий как в реализации, так и по функционалу. Суть технологии вот в следующем: на выход из сети ставится ACL, который выпускает трафик изнутри

наружу. Одновременно с пропуском трафика, автоматически формируется встречный ACL, для пропуска трафика извне вовнутрь. Таким образом появляется возможность получать ответы на свои запросы из интернета.

Приведём пример: есть сеть 192.168.1.0/24 из неё надо организовать доступ в интернет по http, pop и smtp.



Пишется на выход следующие 2 ACL:

```
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
```

Применяем ACL

```
R1(config-if)#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 in
R1(config-if)#
```

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

Предположим, что человек 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 по первой строчке проходит, одновременно в ACL BACK-WWW автоматически на время добавляется зеркальная запись:

```
permit tcp host 123.123.123.123 eq 80 host 192.168.1.100 eq 12345
```

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 established пользуется только флагом в TCP сегменте, а Reflexive реально отслеживает соединения. Флаг можно подделать, в этом случае входящий трафик начнёт 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 справляются с этими задачами отлично.

1.3. 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 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
```

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 standard|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.

1.3.1. Пример именованного списка доступа

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

```
interface Ethernet0/5
ip address 2.0.5.1 255.255.255.0
ip access-group Internet_filter out
ip access-group marketing_group in
ip access-list standard Internet_filter
permit 1.2.3.4
deny any
ip access-list extended marketing_group
permit tcp any 171.69.0.0 0.0.255.255 eq telnet
deny tcp any any
permit icmp any any
deny udp any 171.69.0.0 0.0.255.255 lt 1024
deny ip any any log
```

1.3.2. Ограничение доступа к VTY при помощи ACL

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

Сначала создается стандартный ACL, в котором перечисляем адреса и сети, из который доступ по telnet надо разрешить. Теперь его необходимо применить непосредственно на `line vty 0 4`, то есть, на линии виртульного 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`.

2. Задания для самостоятельного выполнения

В соответствии с Вашим вариантом курсового проекта сконфигурируйте политики безопасности на маршрутизаторе.

3. Содержание отчета

1. Титульный лист.
2. Исходные данные в соответствии с индивидуальным вариантом.
3. Описание всех использованных команд.
4. Скриншот получившихся топологий.
5. Выводы.