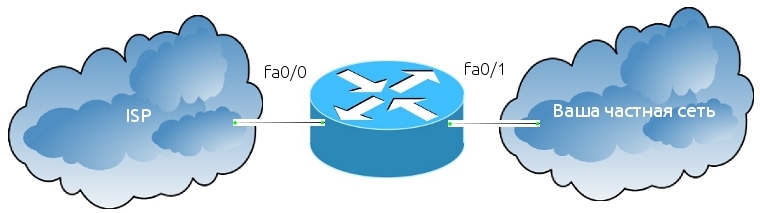
**Введение**

ACL (Access Control List) — это набор текстовых выражений, которые что-то разрешают, либо что-то запрещают. Обычно ACL разрешает или запрещает IP-пакеты, но помимо всего прочего он может заглядывать внутрь IP-пакета, просматривать тип пакета, TCP и UDP порты. Также ACL существует для различных сетевых протоколов (IP, IPX, AppleTalk и так далее). В основном применение списков доступа рассматривают с точки зрения пакетной фильтрации, то есть пакетная фильтрация необходима в тех ситуациях, когда у вас стоит оборудование на границе Интернет и вашей частной сети и нужно отфильтровать ненужный трафик.   
Вы размещаете ACL на входящем направлении и блокируете избыточные виды трафика.

**Теория**

Функционал ACL состоит в классификации трафика, нужно его проверить сначала, а потом что-то с ним сделать в зависимости от того, куда ACL применяется. ACL применяется везде, например:

* На интерфейсе: *пакетная фильтрация*
* На линии Telnet: *ограничения доступа к маршрутизатору*
* VPN: *какой трафик нужно шифровать*
* QoS: *какой трафик обрабатывать приоритетнее*
* NAT: *какие адреса транслировать*

Для применения ACL для всех этих компонентов нужно понять как они работают. И мы в первую очередь будем касаться пакетной фильтрации. Применительно к пакетной фильтрации, ACL размещаются на интерфейсах, сами они создаются независимо, а уже потом они прикручиваются к интерфейсу. Как только вы его прикрутили к интерфейсу маршрутизатор начинает просматривать трафик. Маршрутизатор рассматривает трафик как входящий и исходящий. Тот трафик, который входит в маршрутизатор называется входящим, тот который из него выходит — исходящий. Соответственно ACL размещаются на входящем или на исходящем направлении.  
  
Из вашей частной сети приходит пакет на интерфейс маршрутизатора fa0/1, маршрутизатор проверяет есть ли ACL на интерфейсе или нет, если он есть, то дальше обработка ведется по правилам списка доступа **строго в том порядке, в котором записаны выражения**, если список доступа разрешает проходить пакету, то в данном случае маршрутизатор отправляет пакет провайдеру через интерфейс fa0/0, если список доступа не разрешает проходить пакету, пакет уничтожается. Если списка доступа нет — пакет пролетает без всяких ограничений. Перед тем как отправить пакет провайдеру, маршрутизатор ещё проверяет интерфейс fa0/0 на наличие исходящего ACL. Дело в том, что ACL может быть прикреплен на интерфейсе как входящий или исходящий. К примеру у нас есть ACL с правилом запретить всем узлам в Интернете посылать в нашу сеть пакеты.  
Так на какой интерфейс прикрепить данную ACL? Если мы прикрепим ACL на интерфейс fa0/1 как исходящий, это будет не совсем верно, хотя и ACL работать будет. На маршрутизатор приходит эхо-запрос для какого-то узла в частной сети, он проверяет на интерфейсе fa0/0 есть ли ACL, его нет, дальше проверяет интерфейс fa0/1, на данном интерфейсе есть ACL, он настроен как исходящий, всё верно пакет не проникает в сеть, а уничтожается маршрутизатором. Но если мы прикрепим ACL за интерфейсом fa0/0 как входящий, то пакет будет уничтожатся сразу как пришел на маршрутизатор. Последнее решение является правильным, так как маршрутизатор меньше нагружает свои вычислительные ресурсы. **Расширенные ACL нужно размещать как можно ближе к источнику, стандартные же как можно ближе к получателю**. Это нужно для того, чтобы не гонять пакеты по всей сети зря.  
  
Сам же ACL представляет собой набор текстовых выражений, в которых написано **permit** (разрешить) либо **deny** (запретить), и обработка ведется строго в том порядке в котором заданы выражения. Соответственно когда пакет попадает на интерфейс он проверяется на первое условие, если первое условие совпадает с пакетом, дальнейшая его обработка прекращается. Пакет либо перейдет дальше, либо уничтожится.   
Ещё раз, **если пакет совпал с условием, дальше он не обрабатывается**. Если первое условие не совпало, идет обработка второго условия, если оно совпало, обработка прекращается, если нет, идет обработка третьего условия и так дальше пока не проверятся все условия, **если никакое из условий не совпадает, пакет просто уничтожается**. Помните, в каждом конце списка стоит неявный deny any (запретить весь трафик). Будьте очень внимательны с этими правилами, которые я выделил, потому что очень часто происходят ошибки при конфигурации.  
  
ACL разделяются на два типа:

* Стандартные (Standard): *могут проверять только адреса источников*
* Расширенные (Extended): *могут проверять адреса источников, а также адреса получателей, в случае IP ещё тип протокола и TCP/UDP порты*

Обозначаются списки доступа либо номерами, либо символьными именами. ACL также используются для разных сетевых протоколов. Мы в свою очередь будем работать с IP. Обозначаются они следующим образом, нумерованные списки доступа:

* Стандартные: *от 1 до 99*
* Расширенные: *от 100 до 199*

Символьные ACL разделяются тоже на стандартные и расширенные. Расширенные напомню могут проверять гораздо больше, нежели стандартные, но и работают они медленнее, так как придется заглядывать внутрь пакета, в отличии от стандартных где мы смотрим только поле Source Address (Адрес отправителя). При создании ACL каждая запись списка доступа обозначается порядковым номером, по умолчанию в рамках десяти (10, 20, 30 и т.д). Благодаря чему, можно удалить конкретную запись и на её место вставить другую, но эта возможность появилась в Cisco IOS 12.3, до 12.3 приходилось ACL удалять, а потом создать заново полностью. **Нельзя разместить более 1 списка доступа на интерфейс, на протокол, на направление**. Объясняю: если у нас есть маршрутизатор и у него есть интерфейс, мы можем на входящее направление для IP-протокола разместить только один список доступа, например под номером 10. Ещё одно правило, касающееся самих маршрутизаторов, **ACL не действует на трафик, сгенерированный самим маршрутизатором**.   
Для фильтрации адресов в ACL используется WildCard-маска. Это обратная маска. Берем шаблонное выражение: 255.255.255.255 и отнимаем от шаблона обычную маску.  
255.255.255.255-255.255.255.0, у нас получается маска 0.0.0.255, что является обычной маски 255.255.255.0, только 0.0.0.255 является WildCard маской.

**Виды ACL**

**Динамический (Dynamic ACL)**

Позволяет сделать следующее, например у вас есть маршрутизатор, который подключен к какому-то серверу и нам нужно закрыть доступ к нему из внешнего мира, но в тоже время есть несколько человек, которые могут подключаться к серверу.  
Мы настраиваем динамический список доступа, прикрепляем его на входящем направлении, а дальше людям, которым нужно подключиться, подключаться через Telnet к данному устройству, в результате динамический ACL открывает проход к серверу, и уже человек может зайти скажем через HTTP попасть на сервер. По умолчанию через 10 минут этот проход закрывается и пользователь вынужден ещё раз выполнить Telnet чтобы подключиться к устройству.

**Рефлексивный (Reflexive ACL)**

Здесь ситуация немножко отличается, когда узел в локальной сети отправляет TCP запрос в Интернет, у нас должен быть открытый проход, чтобы пришел TCP ответ для установки соединения. Если прохода не будет — мы не сможем установить соединение, и вот этим проходом могут воспользоваться злоумышленники, например проникнуть в сеть. Рефлексивные ACL работают таким образом, блокируется полностью доступ (deny any) но формируется ещё один специальный ACL, который может читать параметры пользовательских сессий, которые сгенерированны из локальной сети и для них открывать проход в deny any, в результате получается что из Интернета не смогут установить соединение. А на сессии сгенерированны из локальной сети будут приходить ответы.

**Ограничение по времени (Time-based ACL)**

Обычный ACL, но с ограничением по времени, вы можете ввести специальное расписание, которое активирует ту или иную запись списка доступа. И сделать такой фокус, например пишем список доступа, в котором запрещаем HTTP-доступ в течении рабочего дня и вешаем его на интерфейс маршрутизатора, то есть, сотрудники предприятия пришли на работу, им закрывается HTTP-доступ, рабочий день закончился, HTTP-доступ открывается,  
пожалуйста, если хотите — сидите в Интернете.

**Настройка**

Сами ACL создаются отдельно, то есть это просто некий список, который создается в глобальном конфиге, потом он присваивается к интерфейсу и только тогда он и начинает работать. Необходимо помнить некоторые моменты, для того, чтобы правильно настроить списки доступа:

* Обработка ведется строго в том порядке, в котором записаны условия
* Если пакет совпал с условием, дальше он не обрабатывается
* В конце каждого списка доступа стоит неявный deny any (запретить всё)
* Расширенные ACL нужно размещать как можно ближе к источнику, стандартные же как можно ближе к получателю
* Нельзя разместить более 1 списка доступа на интерфейс, на протокол, на направление
* ACL не действует на трафик, сгенерированный самим маршрутизатором
* Для фильтрации адресов используется WildCard маска

**Стандартный список доступа**

Router(config)#**access-list** <номер списка от 1 до 99> **{permit | deny | remark}** **{address | any | host}** [source-wildcard] **[log]**

* permit: *разрешить*
* deny: *запретить*
* remark: *комментарий о списке доступа*
* address: *запрещаем или разрешаем сеть*
* any: *разрешаем или запрещаем всё*
* host: *разрешаем или запрещаем хосту*
* source-wildcard: *WildCard маска сети*
* log: *включаем логгирование пакеты проходящие через данную запись ACL*

**Расширенный список доступа**

Router(config)#**access-list** <номер списка от 100 до 199> **{permit | deny | remark}** protocol source [source-wildcard] [**operator** operand] [**port** <порт или название протокола> **[established]**

* protocol source: *какой протокол будем разрешать или закрывать (ICMP, TCP, UDP, IP, OSPF и т.д)*
* deny: *запретить*
* operator:  *A.B.C.D — адрес получателя  
  any — любой конечный хост  
  eq — только пакеты на этом порте  
  gt — только пакеты с большим номером порта  
  host — единственный конечный хост  
  lt — только пакеты с более низким номером порта  
  neq — только пакеты не на данном номере порта  
  range — диапазон портов*
* port: *номер порта (TCP или UDP), можно указать имя*
* established: *разрешаем прохождение TCP-сегментов, которые являются частью уже созданной TCP-сессии*

**Прикрепляем к интерфейсу**

Router(config-if)#**ip access-group** <номер списка или имя ACL> **{in | out}**

* in: *входящее направление*
* out: *исходящее направление*

**Именованные списки доступа**

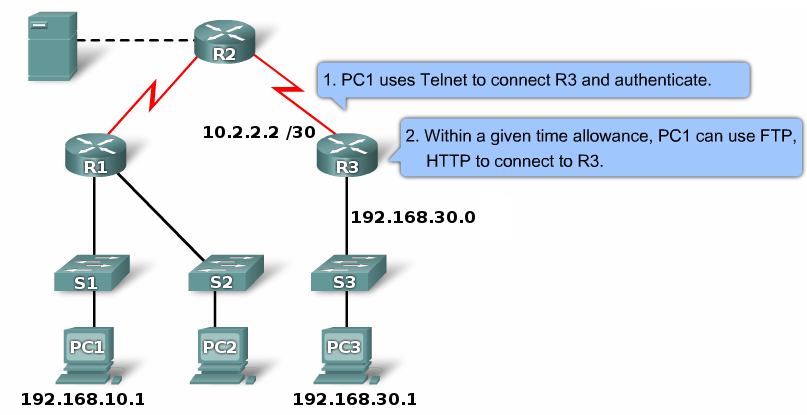
Router(config)#**ip access-list** **{standard | extended} {<номер ACL> | <имя ACL>}**  
Router(config-ext-nacl)# **{default | deny | exit | no | permit | remark}**

* standard: *стандартный ACL*
* extended: *расширенный ACL*
* default: *установить команду в значение по умолчанию*

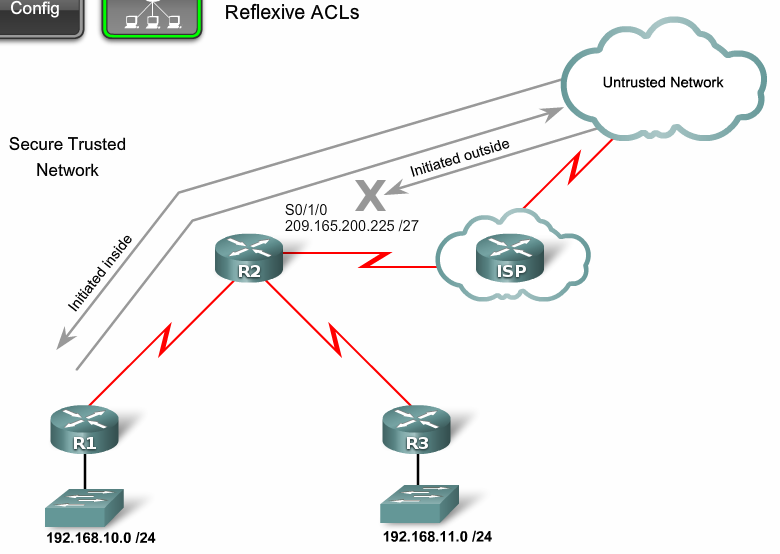
**Ограничение доступа к маршрутизатору**

R(config)#**line vty** 0 4 — переходим в режим настройки виртуальных линий.  
R(config-line)#**password** <пароль>   
R(config-line)#**login**  
R(config-line)#**access-class** 21 **in** — настраиваем логин и пароль, а также закрепляем список доступа с разрешенными IP-адресами.

**Динамические списки доступа**

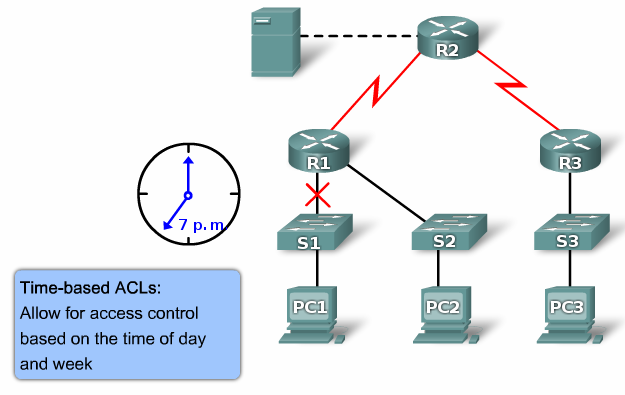
  
  
R3(config)#**username** Student **password** **0** cisco — создаем пользователей для подключения через Telnet.  
R3(config)#**access-list** 101 **permit tcp any host** 10.2.2.2 **eq telnet**  
R3(config)#**access-list** 101 **dynamic testlist timeout 15 permit ip** 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255 — разрешаем подключаться к серверу по Telnet всем узлам.  
R3(config)#interface serial 0/0/1  
R3(config-if)#**ip access-group** 101 in — закрепляем 101 ACL за интерфейсом в входящем направлении.  
R3(config)#**line vty** 0 4  
R3(config-line)#**login local**  
R3(config-line)#**autocommand access-enable host timeout** 5 — как только пользователь аутентифицируеться, сеть 192.168.30.0 будет доступна, через 5 минут бездействия сеанс закроется.

**Рефлексивные списки доступа**



R2(config)#**ip access-list extended** OUTBOUNDFILTERS  
R2(config-ext-nacl)#**permit tcp** 192.168.0.0 0.0.255.255 **any reflect** TCPTRAFFIC  
R2(config-ext-nacl)#**permit icmp** 192.168.0.0 0.0.255.255 **any reflect** ICMPTRAFFIC — заставляем маршрутизатор отслеживать трафик, который инициировался изнутри.  
R2(config)#**ip access-list extended** INBOUNDFILTERS  
R2(config-ext-nacl)#**evaluate** TCPTRAFFIC  
R2(config-ext-nacl)#**evaluate** ICMPTRAFFIC — создаем входящую политику, которая требует, чтобы маршрутизатор проверял входящий трафик, чтобы видеть инициировался ли изнутри и связываем TCPTRAFFIC к INBOUNDFILTERS.  
R2(config)#**interface** serial 0/1/0  
R2(config-if)#**ip access-group** INBOUNDFILTERS in  
R2(config-if)#**ip access-group** OUTBOUNDFILTERS out — применяем входящий и исходящий ACL на интерфейс.

**Ограничение по времени**

  
  
R1(config)#**time-range** EVERYOTHERDAY  
R1(config-time-range)#**periodic** Monday Wednesday Friday 8:00 to 17:00 — создаем список времени, в котором добавляем дни недели и время.  
R1(config)#**access-list** 101 **permit tcp** 192.168.10.0 0.0.0.255 **any eq telnet time-range** EVERYOTHERDAY — применяем time-range к ACL.  
R1(config)#interface s0/0/0  
R1(config-if)#**ip access-group** 101 out — закрепляем ACL за интерфейсом.

**Поиск проблем**

R#**show access-lists** {ACL номер | имя} — смотрим информацию о списке доступа.  
R#**show access-lists** — смотрим все списки доступа на маршрутизаторе.

**Пример**

Router#**show access-lists**   
Extended IP access list **nick**  
**permit ip host 172.168.1.1 host 10.0.0.5**  
**deny ip any any (16 match(es))**  
Standard IP access list **nick5**  
**permit 172.16.0.0 0.0.255.255**  
  
Мы видим что у нас есть два ACL (стандартный и расширенный) под названиями nick и nick5. Первый список разрешает хосту 172.16.1.1 обращаться по IP (это значит что разрешены все протоколы работающие поверх IP) к хосту 10.0.0.5. Весь остальной трафик запрещен показывает команда deny ip any any. Рядом с этим условием в нашем примере пишет (16 match(es)). Это показывает что 16 пакетов попали под это условие.  
Второй ACL разрешает проходить трафик от любого источника в сети 172.16.0.0/16.

# Создание расширенного ACL

**Расширенный ACL** позволяют фильтровать трафик по большому количеству критериев. В этой статье поговорим о настройке расширенного ACL. Рекомендую сначала почитать [общую статью об ACL и статью о](http://ciscotips.ru/acl) [стандартном ACL](http://ciscotips.ru/standard-acl), если вы этого ещё не сделали.

Итак, расширенный ACL может быть именованный и нумерованный. В любом случае, он позволяет фильтровать трафик по следующим параметрам:

1. Адрес отправителя
2. Адрес получателя
3. TCP/UDP порт отправителя
4. TCP/UDP порт получателя
5. Протоколу, завёрнутому в ip (отфильтровать только tcp, только udp, только icmp, только gre и т.п.)
6. Типу трафика для данного протокола (например, для icmp отфильтровать только icmp-reply).
7. Отделить TCP трафик, идущий в рамках установленной TCP сессии от TCP сегментов, которые только устанавливают соединение. Подробнее об этом можно прочитать в статье «[Что делает established в ACL](http://ciscotips.ru/established)»
8. И др.

## Нумерованный расширенный ACL

Нумерованный расширенный ACL должен иметь номер с 100-го по 199 или с 2000 по 2699. Причём номер относится ко всему ACL (ко всем его строчкам).

Например, чтобы запретить web трафик с адреса 192.168.0.1 надо написать следующий стандартный ACL:

R1#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#access-list 110 deny tcp host 192.168.0.1 any eq 80

R1(config)#access-list 110 permit ip any any

Первая строчка запрещает tcp-трафик с указанного адреса на 80-ый порт, вторая – разрешает остальной трафик. access-list 110 – обозначает принадлежность строчки к ACL с номером 110, далее идёт действие (permit – разрешить, deny – запретить, либо remark – комментарий «для себя»). Потом идёт протокол (в CCNA рассматриваются только три протокола tcp, udp и ip). Потом идёт информация об отправителе пакета (адрес, возможно, wildcard маска, возможно, порт отправителя), за тем – то же самое про получателя. В нашем примерев первой строчки порт отправителя не задан, то есть трафик, который будет запрещён может отправляться с любого порта отправителя, порт же получателя должен быть 80.



Применяется такой ACL на интерфейс следующим образом:

R1(config)#interface fa0/0

R1(config-if)#ip access-group 110 in

В этом примере мы применили ACL 110 на интерфейс Fa0/0 для фильтрации входящего трафика.

Нумерованный ACL нельзя отредактировать можно только добавить строчку в конец. В нашем примере добавлять строчку в конец не имеет смысла, так как permit ip any any разрешает весь трафик и дальше проверка не производится. Единственный способ отредактировать нумерованный ACL – вывести кусок конфига с ним, скопировать в блокнот существующий ACL, отредактировать его, удалить старый, вставить новый. Допустим, нам надо добавить в ALC 110 запрет трафика ещё и из второй половины сети 192.168.0.0, с портов выше 1024 на порт 80 . Эта строчка должна встать по логике вещй между первой и второй. Выводим конфиг (весь, или только часть, с acl):

R1#show running-config

…

!

access-list 110 deny tcp host 192.168.0.1 any eq 80

access-list 110 permit ip any any

!

…

Копируем то что касается ACL 110 блокнот, добавляем туда нужную строчку, затем удаляем старый ACL:

R1#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#no access-list 110

И вставляем новый:

R1(config)#access-list 110 deny tcp host 192.168.0.1 any eq 80

R1(config)#access-list 110 deny tcp 192.168.0.128 0.0.0.127 gt 1024 any eq 80

R1(config)#access-list 110 permit ip any any

На всякий случай, повторно применяем на интерфейсе новый ACL

R1(config)#interface fa0/0

R1(config-if)#ip access-group 110 in

## Именованный расширенный ACL

Именованные ACL создаются иначе.

R1(config)#

R1(config)#ip access-list extended MY\_ACL

R1(config-ext-nacl)#deny tcp host 192.168.0.1 any eq 80

R1(config-ext-nacl)#permit ip any any

R1(config-ext-nacl)#end

%SYS-5-CONFIG\_I: Configured from console by console

Мы создали ACL с именем MY\_ACL, который делает то же самое, что и ACL 110 из предыдущего примера. В качестве аргумента команды ip access-list необходимо указать не только имя, но и тип ACL (standard – стандартный и extended – расширенный), так как маршрутизатор не может сам определить тип. В случае использования нумерованного, тип определялся по номеру (с 100-го по 199 или с 2000 по 2699). Внутри именованного ACL уже не надо упоминать имени или номера, так как мы попадаем в режим редактирования конкретного ACL (config-ext-nacl) и все строчки ACL вводим сразу начиная со слова deny, permit или remark.

Применяется именованный ACL так же как и нумерованный:

R1(config)#interface fa0/0

R1(config-if)#ip access-group MY\_ACL in

В отличие от нумерованного, именованный ACL можно редактировать построчно каждой строчке именованного ACL назначается номер со сдвигом 10, то есть строки нумеруются 10,20,30,40, и т.п. Благодаря этому, можно вставить строчку между существующими. Например, если мы зайдём в редактирования ACL MY\_ACL:

R1(config)#ip access-list extended MY\_ACL

И добавим очередную строчку начиная с числа 15

R1(config-ext-nacl)#15 deny tcp host 192.168.0.2 any eq 80

То такая строчка вставится между 10 и 20-ой (между первой и второй) строками старого ACL.

Кроме того, можно удалять строчки по одной. Для этого так же необходимо зайти в режим редактирование ACL и воспользоваться командой no с номером удаляемой строки:

R1(config)#ip access-list extended MY\_ACL

R1(config-ext-nacl)#no 15

R1(config-ext-nacl)#no 10

Чтобы удалить весь ACL, надо выполнить команду:

R1(config)#no ip access-list extended MY\_ACL