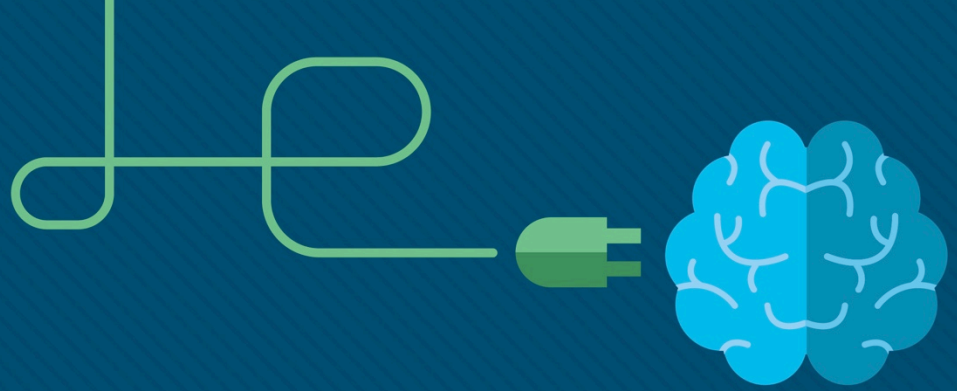


Списки контроля доступа (ACL)

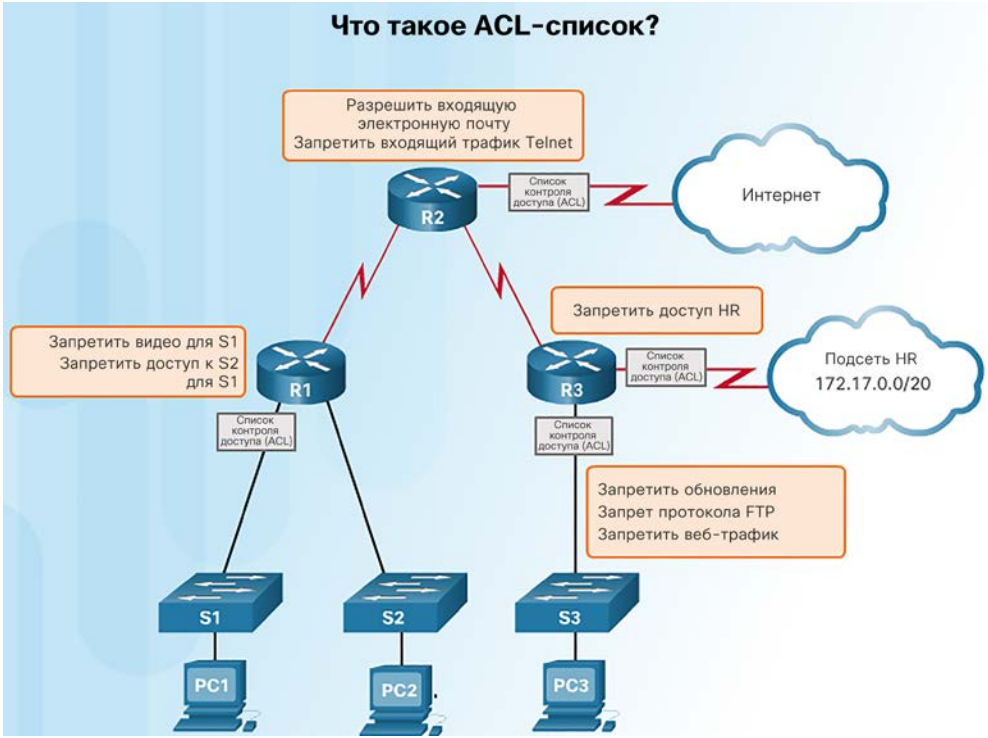
к.т.н., доц. С.А.Зуев

По материалам курса CISCO

CCNA Routing and Switching

Scaling Networks v6.0

Что такое список контроля доступа?



- ACL-список — это ряд команд IOS, определяющих, пересылает ли маршрутизатор пакеты или сбрасывает их, исходя из информации в заголовке пакета. По умолчанию на маршрутизаторе списки контроля доступа не настроены.
- Списки контроля доступа могут выполнять следующие задачи.
  - Ограничение сетевого трафика для повышения производительности сети. Например, можно заблокировать видеотрафик, если он недопустим.
  - Управление потоком трафика. Списки контроля доступа позволяют обеспечить получение обновлений маршрутизации только из известных источников.
  - Списки контроля доступа обеспечивают безопасность сетевого доступа и позволяют блокировать хост или сеть.
  - Фильтрация трафика на основе типа трафика, такого как трафик Telnet.
  - Проверка узлов для разрешения или запрета доступа к сетевым сервисам, таким как FTP или HTTP.

Фильтрация пакетов

- Список контроля доступа (ACL) — это последовательный список разрешающих или запрещающих операторов, называемых записями списка контроля доступа (ACE).
  - Записи списка контроля доступа обычно называют утверждениями списка контроля доступа.
- При прохождении сетевого трафика через интерфейс, где действует список контроля доступа (ACL), маршрутизатор последовательно сопоставляет информацию из пакета с каждой записью в списке контроля доступа на предмет соответствия. Это называется фильтрацией пакетов.
- Фильтрация пакетов:
  - позволяет анализировать входящие и исходящие пакеты;
  - может выполняться на уровне 3 или 4.
- Последняя запись в списке контроля доступа всегда содержит косвенный запрет трафика. Она автоматически вставляется в конец каждого списка контроля доступа и блокирует весь трафик. Поэтому в каждом списке контроля доступа должно быть хотя бы одно разрешающее утверждение.

Принципы работы списков контроля доступа

- Списки контроля доступа не применяются к пакетам, созданным маршрутизатором.
  - Списки контроля доступа определяют набор правил, обеспечивающих дополнительный контроль над пакетами, которые принимаются интерфейсами, транзитными пакетами, которые передаются через маршрутизатор, а также пакетами, которые отправляются из интерфейсов маршрутизатора.
- Списки контроля доступа можно настроить для применения к входящему трафику и к исходящему трафику.
  - Входящие ACL — входящие пакеты обрабатываются перед отправкой в выходной интерфейс.
  - Исходящие ACL — входящие пакеты направляются в выходной интерфейс, а затем обрабатываются исходящим ACL.

Знакомство с шаблонными масками списков контроля доступа

- Списки контроля доступа IPv4 используют шаблонные маски.
- Шаблонная маска — это строка из 32 двоичных цифр (1 и 0), используемая маршрутизатором для определения битов адреса, которые будут рассматриваться на предмет совпадения.
- Шаблонные маски часто называют обратными масками, так как в отличие от маски подсети, где двоичное значение 1 означает совпадение, в шаблонных масках совпадение означает двоичное значение 0. Например:

	Десятичный адрес	Двоичный адрес
IP-адрес для обработки	192.168.10.0	11000000.10101000.00001010.00000000
Шаблонная маска	0.0.255.255	00000000.00000000.11111111.11111111
Итоговый IP-адрес	192.168.0.0	11000000.10101000.00000000.00000000

Примеры шаблонных масок списков контроля доступа

Расчёт шаблонных масок для соответствия узлам и подсетям IPv4

Пример 1

	Десятичные	Двоичные
IP-адрес	192.168.1.1	11000000.10101000.00000001.00000001
Шаблонная маска	0.0.0.0	00000000.00000000.00000000.00000000
Результат	192.168.1.1	11000000.10101000.00000001.00000001

Пример 2

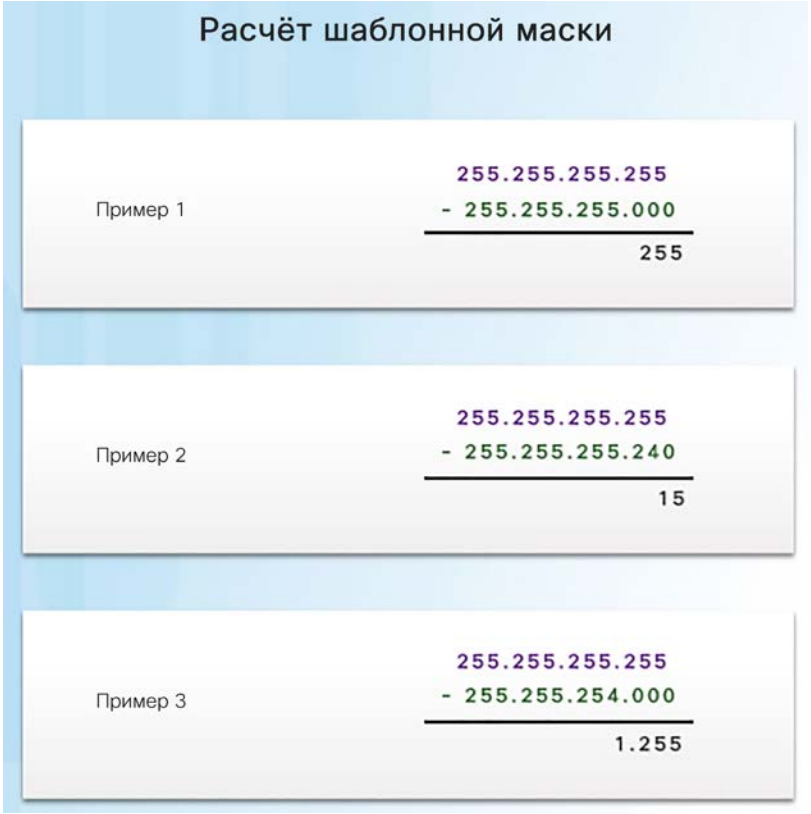
	Десятичные	Двоичные
IP-адрес	192.168.1.1	11000000.10101000.00000001.00000001
Шаблонная маска	255.255.255.255	11111111.11111111.11111111.11111111
Результат	0.0.0.0	00000000.00000000.00000000.00000000

Пример 3

	Десятичные	Двоичные
IP-адрес	192.168.1.1	11000000.10101000.00000001.00000001
Шаблонная маска	0.0.0.255	00000000.00000000.00000000.11111111
Результат	192.168.1.0	11000000.10101000.00000001.00000000

- Чтобы научиться вычислять шаблонную маску, соответствующую подсетям IPV4, требуется практика. В первом слева:
  - Пример 1. Шаблонная маска предусматривает, что каждый бит в адресе IPv4 192.168.1.1 должен точно соответствовать.
  - Пример 2. Шаблонная маска предусматривает, что соответствовать будет все.
  - Пример 3. Шаблонная маска предусматривает, что соответствовать будет любой хост в сети 192.168.1.0/24.

Вычисление шаблонных масок



- Примеры вычисления шаблонных масок:
  - Пример 1. Предположим, требуется разрешить доступ всем пользователям из сети 192.168.3.0 с маской подсети 255.255.255.0. Вычтите подсеть из 255.255.255.255 — в результате получается 0.0.0.255.
  - Пример 2. Предположим, требуется разрешить доступ к сети 14 пользователям из подсети 192.168.3.32/28 с маской подсети 255.255.255.240. После вычитания маски подсети из 255.255.255.255 получается 0.0.0.15.
  - Пример 3. Предположим, что требуется сопоставить только сети 192.168.10.0 и 192.168.11.0 с маской подсети 255.255.254.0. После вычитания маски подсети из 255.255.255.255 получается 0.0.1.255.

Ключевые слова для шаблонных масок

Сокращения шаблонной маски

Пример 1

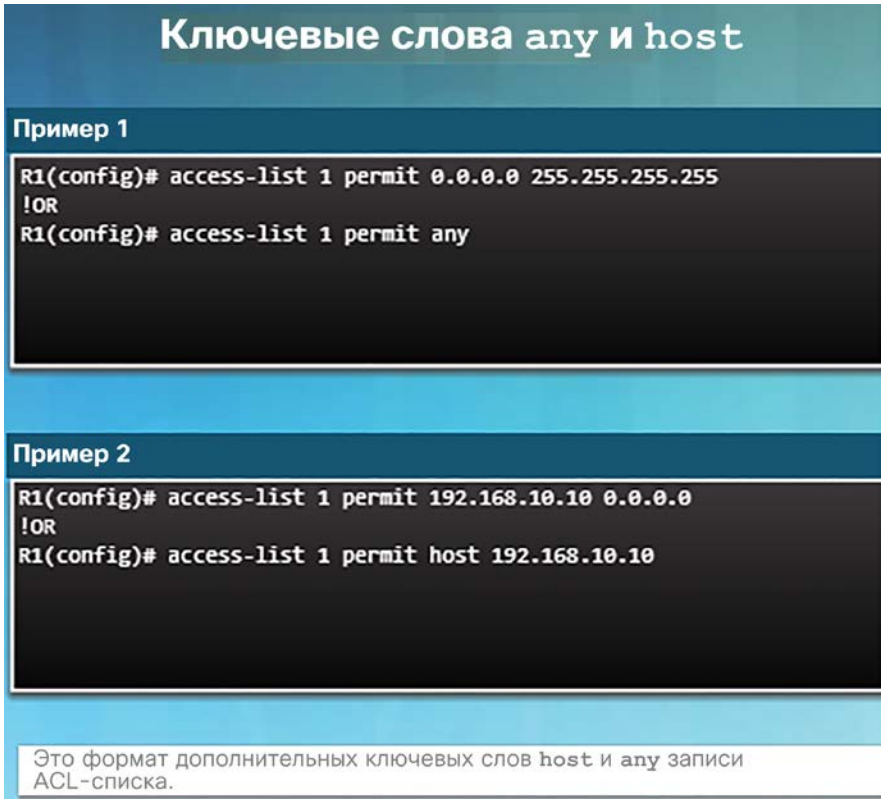
- 192.168.10.10 0.0.0.0 сопоставляет все биты адреса
- Сократите эту шаблонную маску, используя IP-адрес, перед которым указано ключевое слово `host` (`host 192.168.10.10`)

Пример 2

- 0.0.0.0 255.255.255.255 игнорирует все биты адреса
- Это выражение можно сократить с помощью ключевого слова `any`

- Для упрощения чтения шаблонных масок используются ключевые слова **host** и **any**, помогающие определить наиболее распространенные варианты применения шаблонных масок.
  - **host** замещает маску 0.0.0.0
  - **any** замещает маску 255.255.255.255
- Если требуется сопоставить адрес 192.169.10.10, можно использовать выражение **192.168.10.10 0.0.0.0** или **host 192.168.10.10**
- В примере 2 вместо ввода выражения **0.0.0.0 255.255.255.255** можно использовать одно ключевое слово **any**.

Примеры с ключевыми словами для шаблонных масок



- В примере 1 на рисунке иллюстрируется применение ключевого слова **any** вместо адреса IPv4 0.0.0.0 с шаблонной маской 255.255.255.255.
- В примере 2 показывается, как использовать ключевое слово **host** для замены шаблонной маски при определении одного хоста.

Общие рекомендации по созданию списков контроля доступа

- Используйте ACL-списки в межсетевых экранах маршрутизаторов, размещенных между внутренней и внешней сетями, например, Интернетом.
- Используйте списки контроля доступа на маршрутизаторе, расположенном между двумя частями сети, для контроля трафика, входящего или исходящего из определенной части этой внутренней сети.
- Настраивайте списки контроля доступа на граничных маршрутизаторах, например, расположенных на периметре сети. Это обеспечит базовый буфер от внешней сети, которую вы не контролируете.
- Настройте ACL-списки для каждого протокола сети, настроенного на интерфейсе пограничного маршрутизатора.

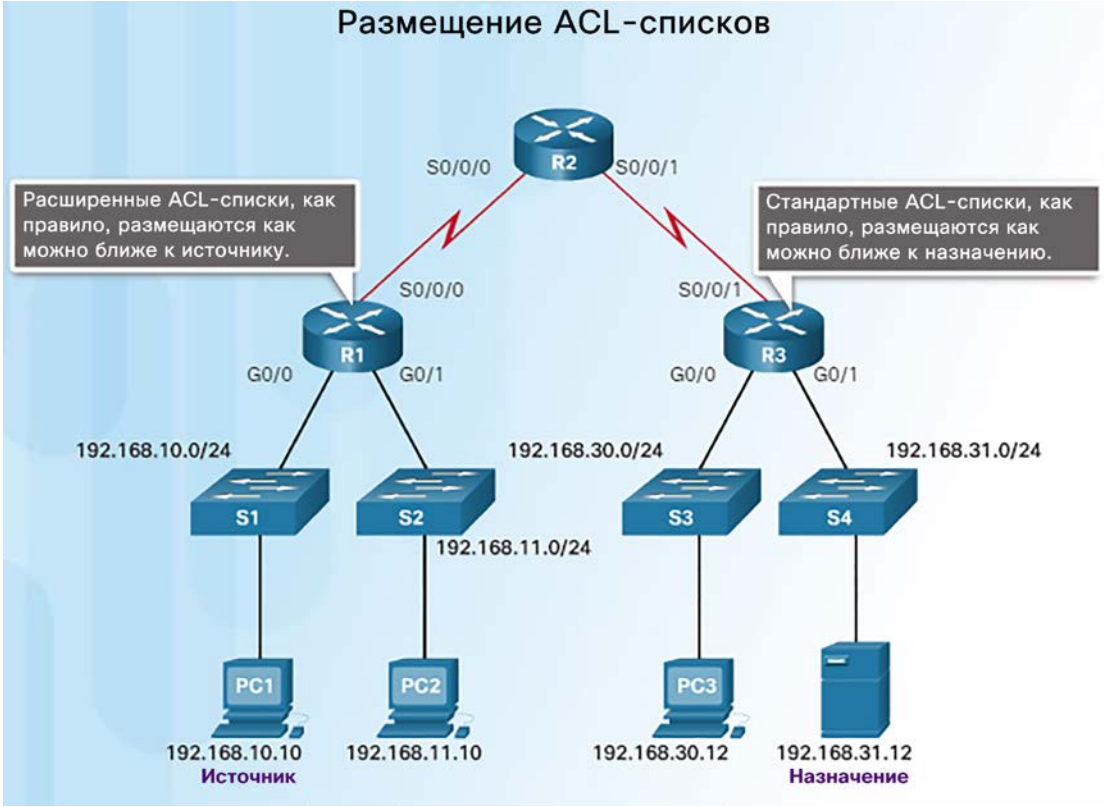
Практические рекомендации по спискам контроля доступа

Рекомендации по созданию ACL-списков

Рекомендации	Преимущество
Создавайте ACL-списки, исходя из корпоративной политики обеспечения информационной безопасности.	Соблюдение рекомендации обеспечивает соответствие требованиям информационной безопасности компании.
Подготовьте описание обязательных действий ваших ACL-списков.	Соблюдение рекомендации поможет избежать непреднамеренного создания потенциальных проблем доступа.
Используйте текстовый редактор для создания, редактирования и сохранения ACL-списков.	Соблюдение рекомендации поможет создать библиотеку повторно используемых ACL-списков.
Проверьте работу ACL-списков в пробной сети перед внедрением в реальную действующую сеть.	Соблюдение рекомендации поможет избежать дорогостоящих ошибок.

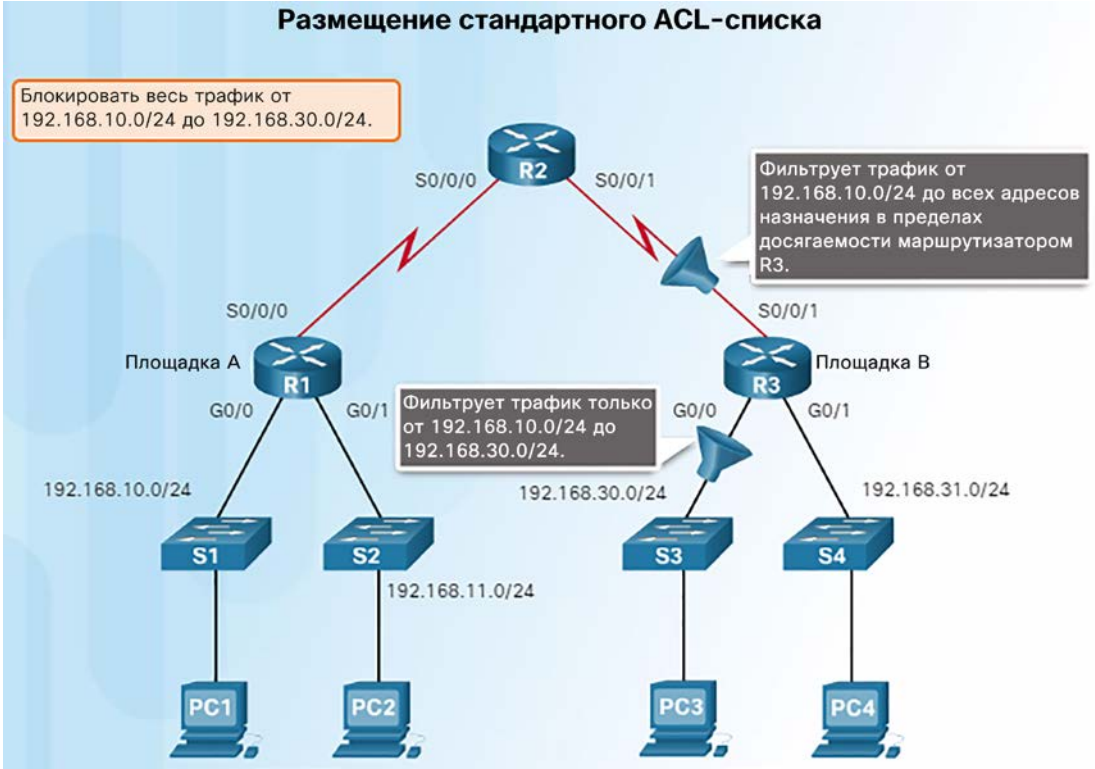
- При использовании списков контроля доступа значительное внимание необходимо уделять деталям. Ошибки могут привести к серьезным последствиям и значительным затратам, связанным с простоями, поиском и устранением неполадок, а также со сниженной производительностью работы сети.

Общие рекомендации по созданию списков контроля доступа



- Правильное размещение ACL-списка может повысить эффективность сети. Например, можно разместить список контроля доступа для сокращения объема ненужного трафика.
- Каждый список контроля доступа (ACL) должен быть размещен там, где он может демонстрировать максимальную эффективность.
  - Расширенные списки контроля доступа. Расширенные списки контроля доступа следует размещать как можно ближе к источнику фильтруемого трафика. Это предотвращает нежелательный трафик максимально близко к источнику без пересечения им сетевой инфраструктуры.
  - Стандартные списки контроля доступа. Так как в стандартных списках контроля доступа не указываются адреса назначения, их следует размещать их как можно ближе к месту назначения.

Размещение стандартных списков контроля доступа



- В этом примере показано правильное размещение стандартного списка контроля доступа, который настроен для блокирования трафика, идущего из сети 192.168.10.0/24 в сеть 192.168.30.0/24.
- Есть два возможных места, в которых можно настроить этот список контроля доступа на маршрутизаторе R3.
- Если этот список контроля доступа применить к интерфейсу S0/0/1, он будет блокировать трафик к сети 192.168.30.0/24**, но также** и к сети 192.168.31.0/24.
- Лучше место для применения этого списка контроля доступа — интерфейс G0/0 маршрутизатора R3. Список контроля доступа следует применить к трафику, исходящему из интерфейса G0/0. Пакеты из сети 192.168.10.0/24 по-прежнему могут достигать сети 192.168.31.0/24.

Синтаксис стандартных нумерованных списков контроля доступа IPv4

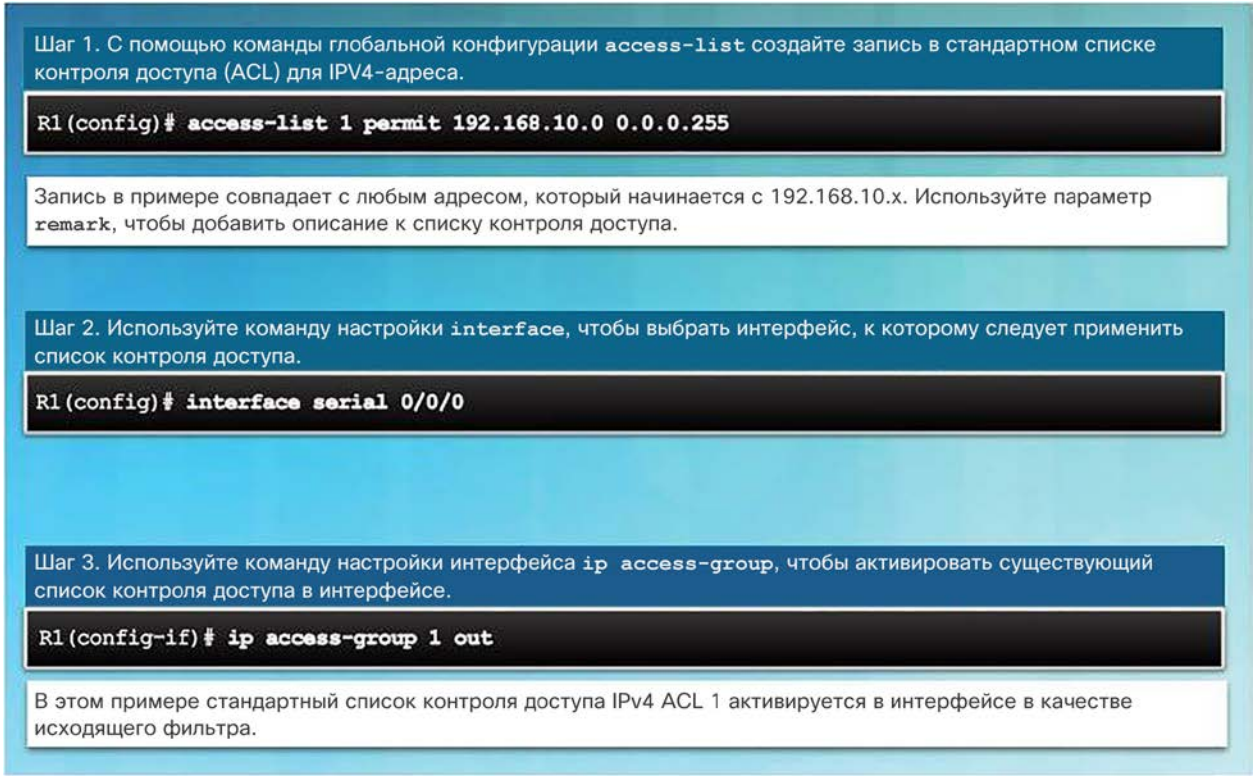
Параметр	Описание
access-list-number	Номер ACL-списка. Это десятичное число от 1 до 99 или от 1300 до 1999 (для стандартного ACL-списка).
deny	Запрещает доступ при совпадении условий.
permit	Разрешает доступ при совпадении условий.
remark	Чтобы сделать список проще для понимания и прочтения, добавьте комментарий о записях в списке доступа IP.
source	Номер сети или узла, с которых отправляется пакет. Два способа определить адрес *источника*: • Используйте 32-битный адрес, записанный в виде четырех 8-битовых целых чисел, разделенных точками. • Используйте ключевое слово **any** как сокращение для адреса ***источника и групповой маски источника*** 0.0.0.0 255.255.255.255.
source-wildcard	(Опционально). 32-битная шаблонная маска должна применяться к адресу источника. Разряды в позиции битов, которые вы хотите игнорировать.
log	(Опционально). Вызывает информационное сообщение журнала о пакете, соответствующем записи, которая должна быть отправлена на консоль. (Уровень сообщений, регистрируемых на консоли, регулируется командой **logging console**). Сообщение включает номер ACL-списка, указание, был ли пакет разрешен или запрещен, адрес источника и количество пакетов. Сообщение создается для первого соответствующего пакета, затем – с пятиминутным интервалом, включая количество разрешенных и запрещенных пакетов в предшествующем пятиминутном интервале.

- Команда глобальной конфигурации **access-list** определяет стандартный ACL-список с номером в диапазоне от 1 до 99.
- Полный синтаксис команды стандартного ACL-списка:

  Router(config)# **access-list** *номер-списка-контроля-доступа* { **deny** | **permit** | **remark** } *источник* [*шаблонная-маска-источника*][**log**]

Для удаления ACL-списка используется команда глобальной конфигурации **no access-list**. Для проверки удаления списка контроля доступа используется команда **show access-list**.

Применение стандартных списков контроля доступа IPv4 к интерфейсам



- Создав стандартный список контроля доступа (ACL), его необходимо связать с интерфейсом при помощи команды **ip access-group**, которая вводится в режиме интерфейсной настройки:

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

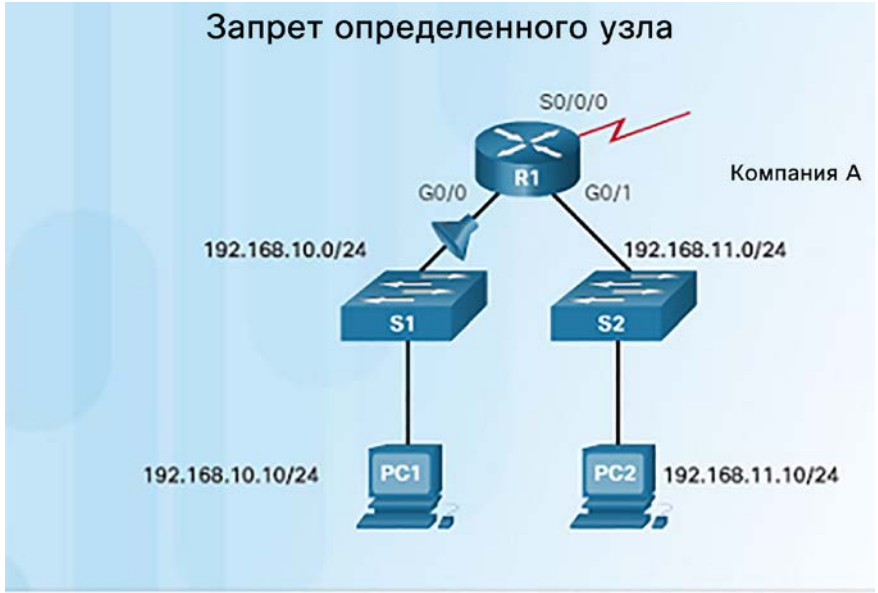
- Для удаления всего списка контроля доступа из интерфейса сначала следует ввести команду **no ip access-group** на интерфейсе, а затем ввести глобальную команду **no access-list**.

Примеры стандартных нумерованных списков контроля доступа IPv4


```
R1(config)# no access-list 1
R1(config)# access-list 1 deny host 192.168.10.10
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 1 out
```

- На рисунке слева показан пример списка контроля доступа, который разрешает трафик от определенной подсети, но запрещает трафик от конкретного хоста из этой подсети.
  - Команда **no access-list 1** удаляет предыдущую версию списка контроля доступа ACL 1.
  - Следующая инструкция списка контроля доступа запрещает хост 192.168.10.10.
  - Каким еще способом можно задать эту команду без использования ключевого слова **host**?
  - Затем разрешаются все остальные узлы из сети 192.168.10.0/24.
  - Здесь присутствует неявный запрет, соответствующий всем остальным сетям.
  - Далее этот список контроля доступа заново применяется к интерфейсу в исходящем направлении.

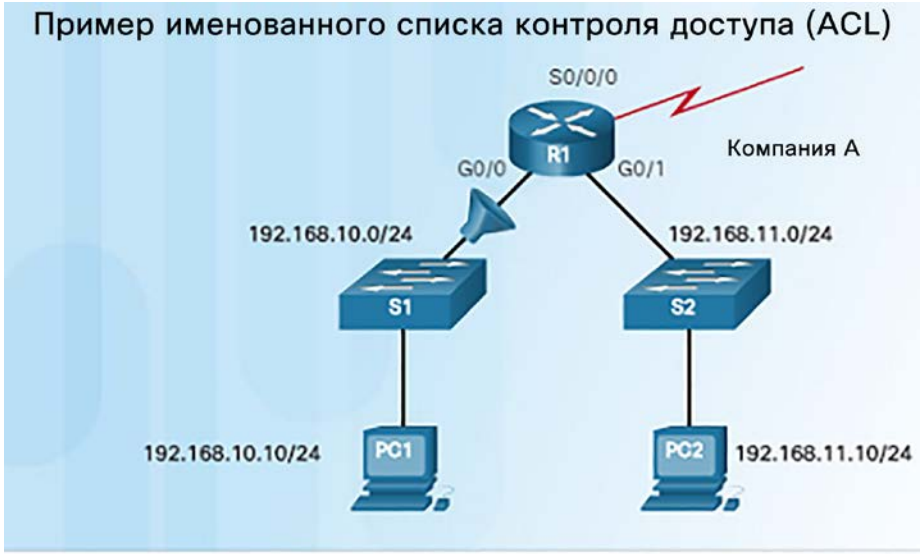
Примеры стандартных нумерованных списков контроля доступа IPv4 (продолжение)




```
R1(config)# no access-list 1
R1(config)# access-list 1 deny host 192.168.10.10
R1(config)# access-list 1 permit any
R1(config)# interface g0/0
R1(config-if)# ip access-group 1 in
```

- В этом примере демонстрируется список контроля доступа, запрещающий определенный хост, но разрешающий весь остальной трафик.
  - Первое утверждение списка контроля доступа удаляет предыдущую версию ACL 1.
  - Следующая команда с помощью ключевого слова deny будет отклонять трафик от хоста PC1, расположенного в сети 192.168.10.10.
  - Утверждение **access-list 1 permit any** будет разрешать все остальные узлы.
  - Этот список контроля доступа применяется к интерфейсу G0/0 во входящем направлении, так как он затрагивает только локальную сеть 192.168.10.0/24.

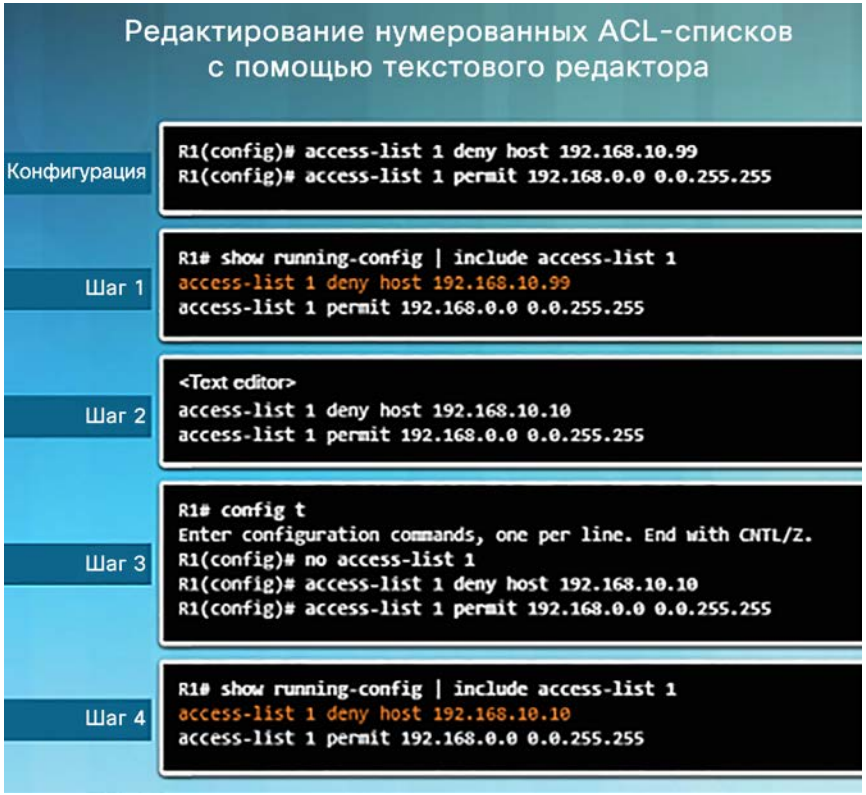
Синтаксис именованных стандартных списков контроля доступа IPv4




```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# permit any
R1(config-std-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group NO_ACCESS out
```

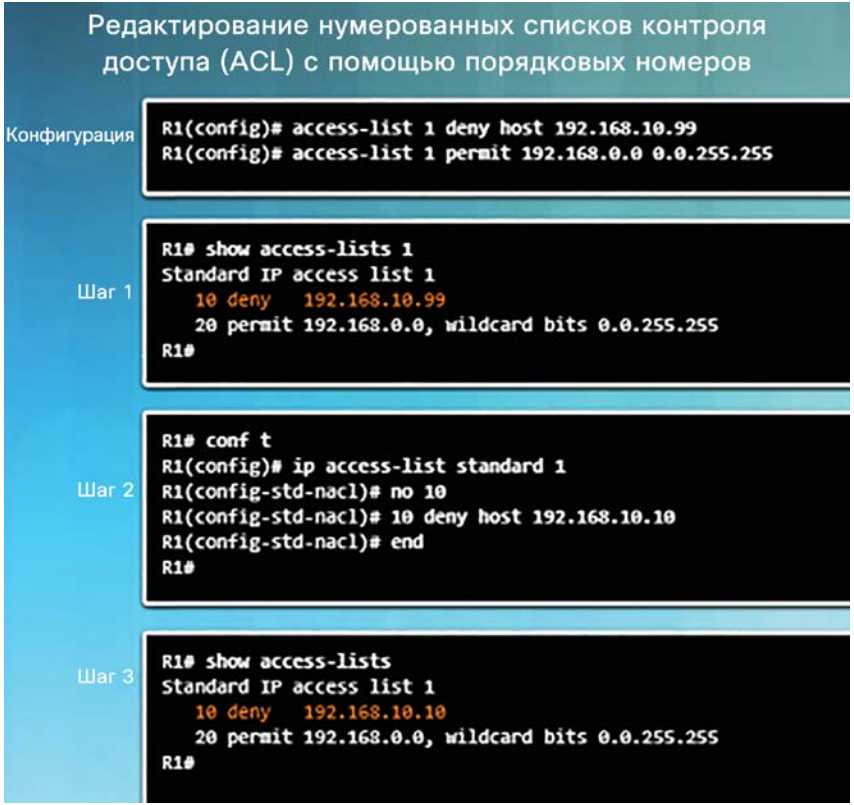
- Обозначение списка контроля доступа по имени, а не по номеру, упрощает понимание его функции.
- В приведенном слева примере показано, как настроить именованный стандартный список контроля доступа. Обратите внимание, что команды немного отличаются.
  - Для создания именованного списка контроля доступа используется команда **ip access-list**. Имена списков контроля доступа состоят из букв и цифр, учитывают регистр и должны быть уникальными.
  - При необходимости используйте утверждение permit или deny. Можно также добавлять комментарии с помощью команды **remark**.
  - Для применения списка контроля доступа к интерфейсу служит команда **ip access-group** *имя*.

Метод 1. Текстовый редактор



- Иногда проще создавать и редактировать списки контроля доступа в текстовом редакторе, например в Блокноте Microsoft, чем вносить изменения непосредственно на маршрутизаторе.
- Если список контроля доступа уже существует, выведите его на экран с помощью команды **show running-config**, скопируйте и вставьте его в текстовый редактор, внесите необходимые изменения, а затем скопируйте и вставьте его обратно в интерфейс маршрутизатора.
- Важно отметить, что при применении команды no access-list разные версии ПО IOS ведут себя по-разному.
  - Если список контроля доступа, который был удален, все еще значится примененным к интерфейсу, одни версии IOS действуют так, будто никакие списки контроля доступа не защищают сеть, в то время как другие версии блокируют весь трафик.

Метод 2. Порядковые номера



- На рисунке слева показаны действия для внесения изменений в нумерованный список контроля доступа с использованием порядковых номеров.
- На шаге 1 выявляется проблема. Утверждение **deny 192.168.10.99** неправильное. Хост, который требуется заблокировать, имеет адрес 192.168.10.10.
- На шаге 2 показано, как перейти в стандартный список контроля доступа 1 и внести изменения. Неправильное утверждение удаляется с помощью команды no: **no 10**
- После удаления добавляется новое утверждение с правильным узлом: **10 deny host 192.168.10.10**

Изменение именованных стандартных списков контроля доступа

```
R1# show access-lists
Standard IP access list NO_ACCESS
    10 deny   192.168.11.10
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# 15 deny host 192.168.11.11
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list NO_ACCESS
    10 deny   192.168.11.10
    15 deny   192.168.11.11
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

- Команда именованного списка контроля доступа **no** *последовательный-номер* используется для удаления отдельных утверждений.

- Используя порядковые номера утверждений, можно легко вставлять или удалять отдельные утверждения.
- На рисунке слева показан пример вставки строки в именованный список контроля доступа.
- Так как эта команда имеет номер 15, она будет помещена между утверждениями 10 и 20.
- Обратите внимание, что при первоначальном создании списка контроля доступа сетевой администратор назначил каждой команде номера с шагом 10, чтобы оставить место для изменений и дополнений.

Проверка списков контроля доступа

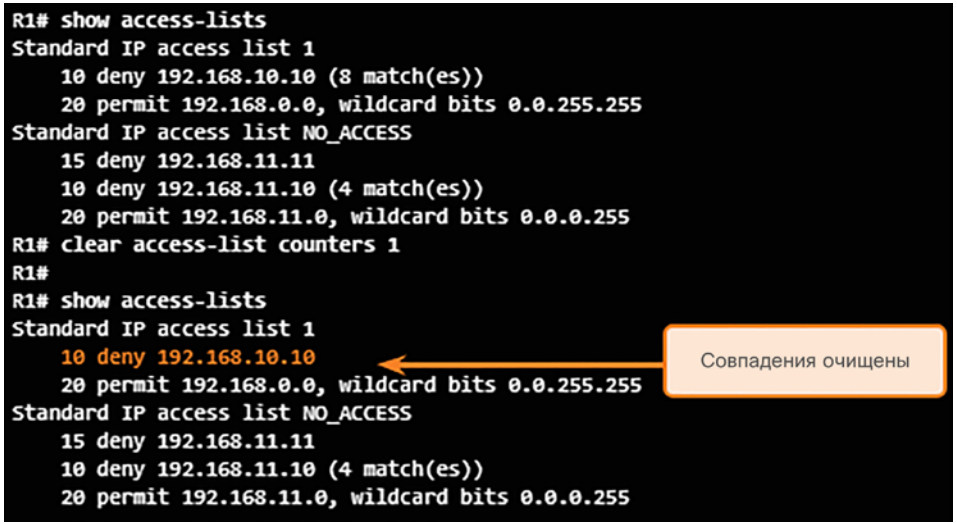
```
R1# show ip interface s0/0/0
Serial0/0/0 is up, line protocol is up
  Internet address is 10.1.1.1/30
<output omitted>
  Outgoing access list is 1
  Inbound access list is not set
<output omitted>

R1# show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
<output omitted>
  Outgoing access list is NO_ACCESS
  Inbound access list is not set
<output omitted>
```

```
R1# show access-lists
Standard IP access list 1
  10 deny 192.168.10.10
  20 permit 192.168.0.0, wildcard bits 0.0.255.255
Standard IP access list NO_ACCESS
  15 deny 192.168.11.11
  10 deny 192.168.11.10
  20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

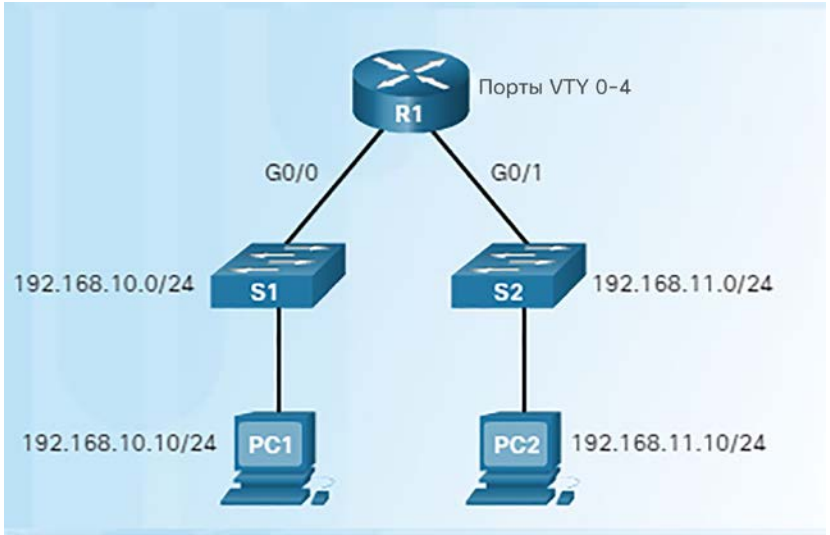
- Для проверки правильности применения списка контроля доступа к интерфейсу используйте команду **show ip interface**.
- В выходных данных этой команды приводится имя списка контроля доступа и направление, в котором он был применен к интерфейсу.
- Чтобы отобразить списки контроля доступа, настроенные на маршрутизаторе, используйте команду **show access-lists**.
- Обратите внимание, что для списка контроля доступа NO_ACCESS утверждения отображаются не в последовательном порядке. Эта ситуация будет рассмотрена далее в этом разделе.

Статистика по ACL



- С помощью команды **show access-lists** можно отобразить соответствующую статистику после применения списка контроля доступа к интерфейсу и выполнения проверки.
- Когда создается трафик, который должен соответствовать какому-либо утверждению списка контроля доступа, количество совпадений, отображаемых в выходных данных команды **show access-lists**, должно увеличиться.
- Напомним, что в каждом списке контроля доступа есть неявное последнее утверждение **deny any**. Статистика для этой неявной команды не отображается. Однако если эта команда настроена вручную, результаты будут отображаться.
- Чтобы очистить счетчики для тестирования, можно использовать команду clear access-list counters.

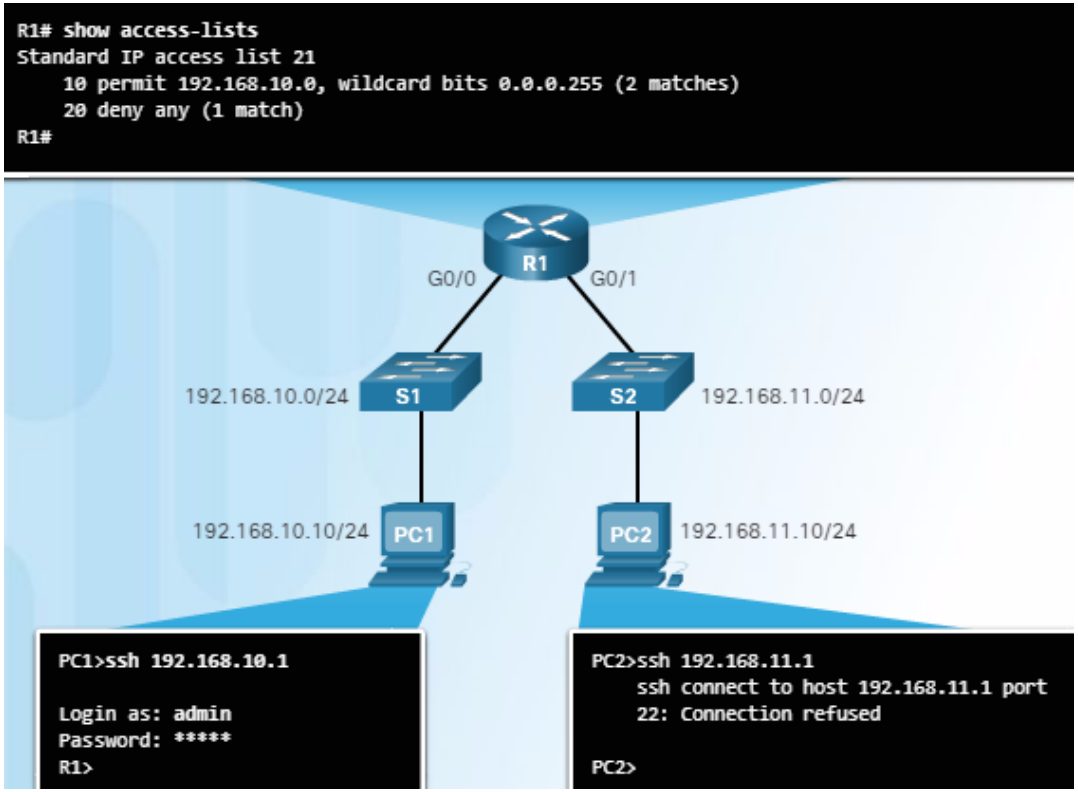
Команда access-class




```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# access-class 21 in
R1(config-line)# exit
R1(config)# access-list 21 permit 192.168.10.0 0.0.0.255
R1(config)# access-list 21 deny any
```

- Для повышения информационной безопасности необходимо ограничивать административный доступ VTY к устройствам Cisco.
- Ограничение доступа VTY позволяет задать перечень IP-адресов, которым разрешен удаленный доступ к процессу EXEC на маршрутизаторе.
- Команда access-class, настроенная в режиме линейного конфигурирования, будет ограничивать входящие и исходящие соединения между конкретным VTY (на устройстве Cisco) и адресами в списке контроля доступа.
- Router(config-line)# **access-class** *номер-списка-контроля-доступа* {in [vrf-also] | out }

Проверка безопасности порта VTY

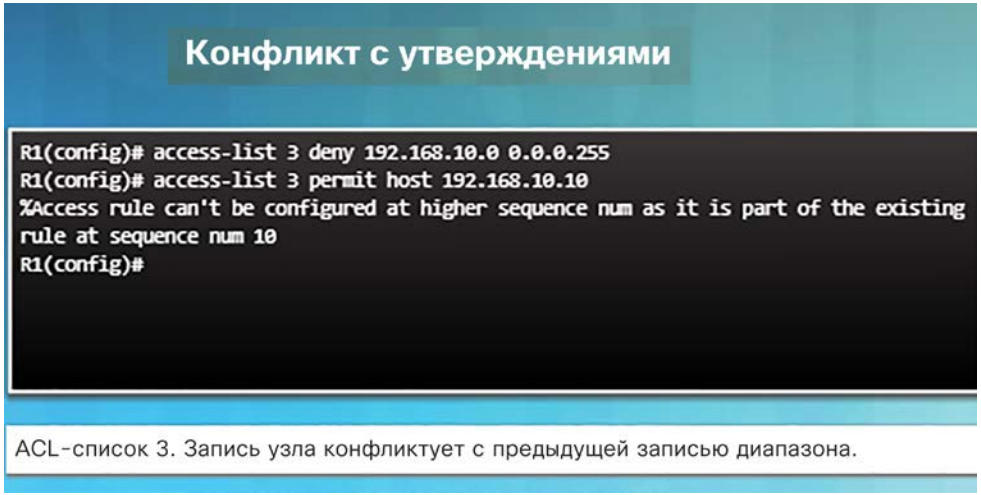


- Проверка конфигурации списка контроля доступа, используемого для ограничения доступа VTY, очень важна.
- На рисунке слева показаны два устройства, пытающиеся установить соединение по протоколу SSH с двумя разными устройствами.
- В выходных данных команды show access-lists показаны результаты после попыток PC1 и PC2 установить соединение SSH.
- Обратите внимание на совпадения в утверждениях permit и deny.

Неявная блокировка трафика (Deny Any)

- Если ACL-список состоит из одной команды запрета, весь трафик будет отклоняться.
- В списке контроля доступа должна быть настроена по крайней мере одна запись разрешения; в противном случае весь трафик будет заблокирован.
- Изучите два списка контроля доступа на рисунке слева.
  - Результаты их применения будут одинаковыми или разными?

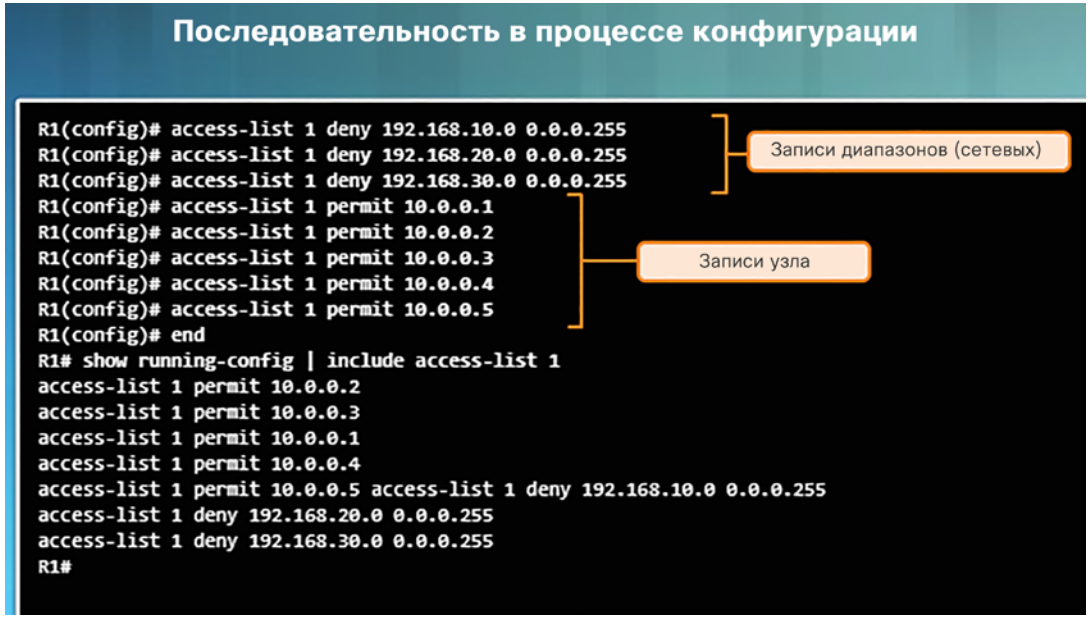
Порядок записей в списке контроля доступа



ACL-список 3. Запись узла конфликтует с предыдущей записью диапазона.

- Порядок записей в списке контроля доступа важен, так как записи списка контроля доступа обрабатываются последовательно.
- На рисунке слева показан конфликт между двумя утверждениями, так как они расположены в неверном порядке.
  - Первое утверждение deny блокирует все из сети 192.168.10.0/24.
  - Однако второе утверждение permit пытается разрешить узел 192.168.10.10.
  - Это утверждение отклоняется, так как узел входит в сеть, указанную в предыдущем утверждении.
  - Чтобы разрешить проблему, достаточно поменять порядок этих двух утверждений.

Изменение порядка стандартных списков контроля доступа в Cisco IOS



- Запишите порядок, в котором вводились утверждения списка контроля доступа во время настройки.
- Отметьте, что после ввода команды **show running-config** утверждения были отображены в другом порядке.
- Утверждения для хостов перечислены первыми, но не в том порядке, в каком они вводились.
- IOS располагает утверждения для хостов с помощью специальной функции хеширования. В результате такой порядок позволяет оптимизировать поиск записи списка контроля доступа для хоста.
- Порядок расположения утверждений для диапазонов остался неизменным. Функция хеширования применяется только к утверждениям для хостов.

Процессы маршрутизации и списки контроля доступа

- На рисунке проиллюстрированы логика работы маршрутизации и процессов ACL-списка.
- При получении пакета в интерфейсе маршрутизатора процесс маршрутизации остается неизменным независимо от того, настроены ли списки контроля доступа.
- После извлечения информации кадра маршрутизатор проверяет наличие списка контроля доступа на входном интерфейсе. При наличии списка контроля доступа пакет сопоставляется с утверждениями списка.
- Если пакет соответствует одной из записей, он принимается или отклоняется — в зависимости от условия, с которым он совпал.
- Если пакет разрешен, то после обработки пакета маршрутизатор ищет список контроля доступа в выходном интерфейсе.

Расширенные ACL

- ACLs numbered1–99 or 1300–1999 are standard IPv4 ACLs.
- Standard ACLs match packets by examining the source IP address field in the IP header of that packet.
- Standard ACLs are used to filter packets based solely on Layer 3 source information.

- Расширенные списки доступа дают возможность более точно фильтровать трафик, поэтому и используются они чаще. Помимо адреса источника они еще проверяют:
  - протокол (IP, ICMP, TCP, UDP и др)
  - адрес назначения
  - номер порта (не интерфейса)

Синтаксис команды расширенного нумерованного ACL

```
access-list { acl-# } { permit | deny | remark } protocol
source-addr [ source-wildcard ]  destination-
addr [destination-wildcard ] [ operator operand ] [port]
[ established ][ log ]
```

	199 (for an extended IP ACL) and 2000 to 2699 (expanded IP ACLs).
deny	Denies access if the conditions are matched.
permit	Permits access if the conditions are matched.
remark	Used to enter a remark or comment.
protocol	Name or number of an Internet protocol. Common keywords include **icmp**, **ip**, **tcp**, or **udp**. To match any Internet protocol (including ICMP, TCP, and UDP) use the **ip** keyword.
source-addr	Number of the network or host from which the packet is being sent.
source-wildcard	Wildcard bits to be applied to source.
destination-addr	Number of the network or host to which the packet is being sent.
destination-wildcard	Wildcard bits to be applied to the destination.
operator	(Optional) Compares source or destination ports. Possible operands include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
port	(Optional) The decimal number or name of a TCP or UDP port.
established	(Optional) For the TCP protocol only: Indicates an established connection.

Синтаксис команды расширенного нумерованного ACL

- **Router(config)#access-list *access-list-number* [deny | permit | remark] *protocol source [source-wildcard]* [operator*operand] [port port-number or name] destination [destination-wildcard]* [operator *operand] [port port-number or name]*[established]**

  где:

  access-list-number - номер ACL (100-199 и 2000-2699)

  deny - запретить трафик

  permit - разрешить трафик

  remark - описание правила или ACL

  protocol - имя или номер протокола. В основном это - IP, ICMP, TCP, UDP.

  source - адрес источника

  source-wildcard - обратная маска источника

  destination - адрес назначения

  destination-wildcard - обратная маска назначения

  operator - сравнивает номера портов. Может быть: lt-меньше чем, gt-больше чем, eq-равно, neq-не равно, **range**-включает диапазон.

  port - номер порта

  established - только для TCP - указывает установленное соединение.

Extended Named IP ACLs

```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
```

The *Log* parameter can be used to log matches to ACLs. The following information is included:

Action - Permit or deny

Protocol - TCP, UDP, or ICMP

Source and destination - IPv4 or IPv6 addresses

TCP and UDP - Source and destination port numbers

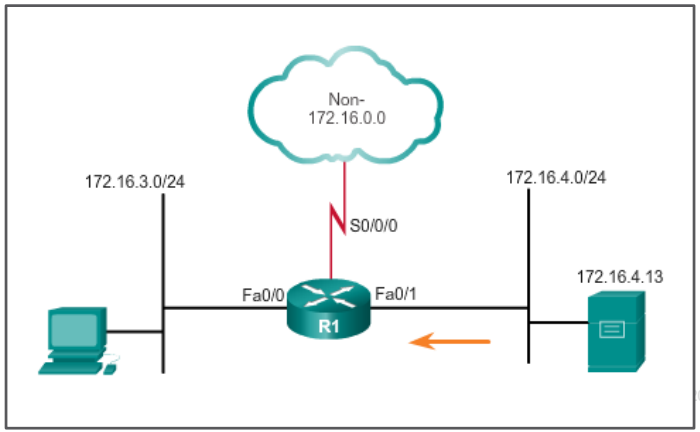
For ICMP - Message types

Log messages are generated on the first packet match and then at five-minute intervals after that first packet match.

Пример расширенного ACL

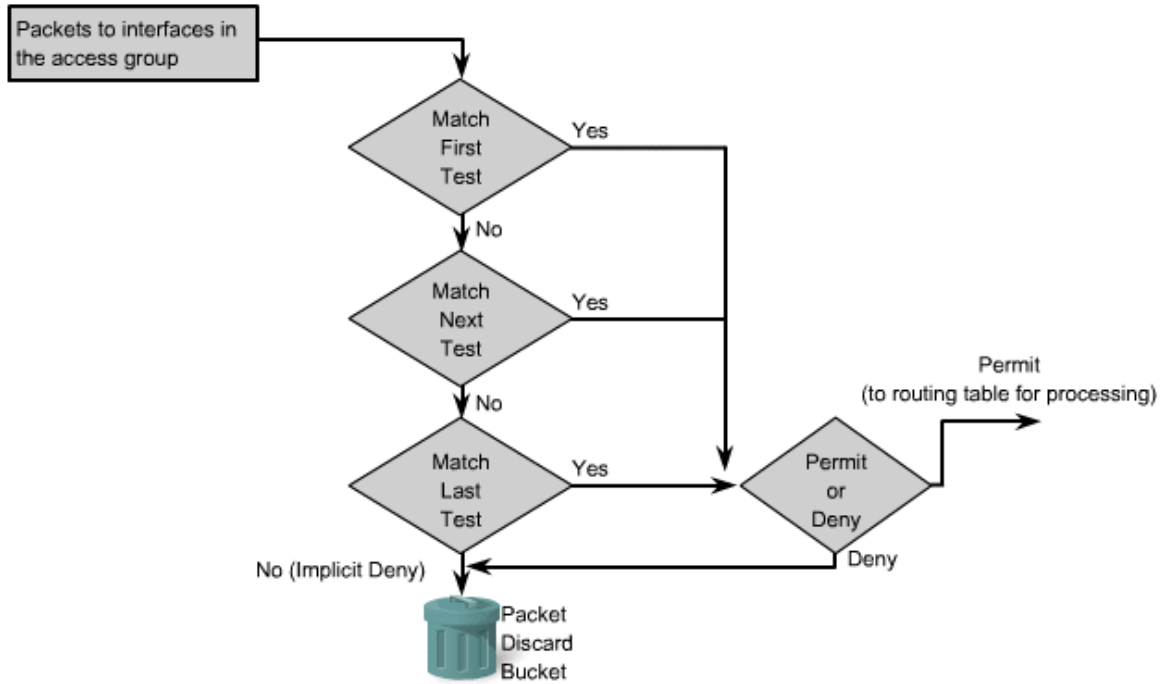
FTP traffic from one subnet must be denied on another subnet.

- `R1(config)# access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 21`
- `R1(config)# access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 20`
- `R1(config)# access-list 101 permit ip any any`



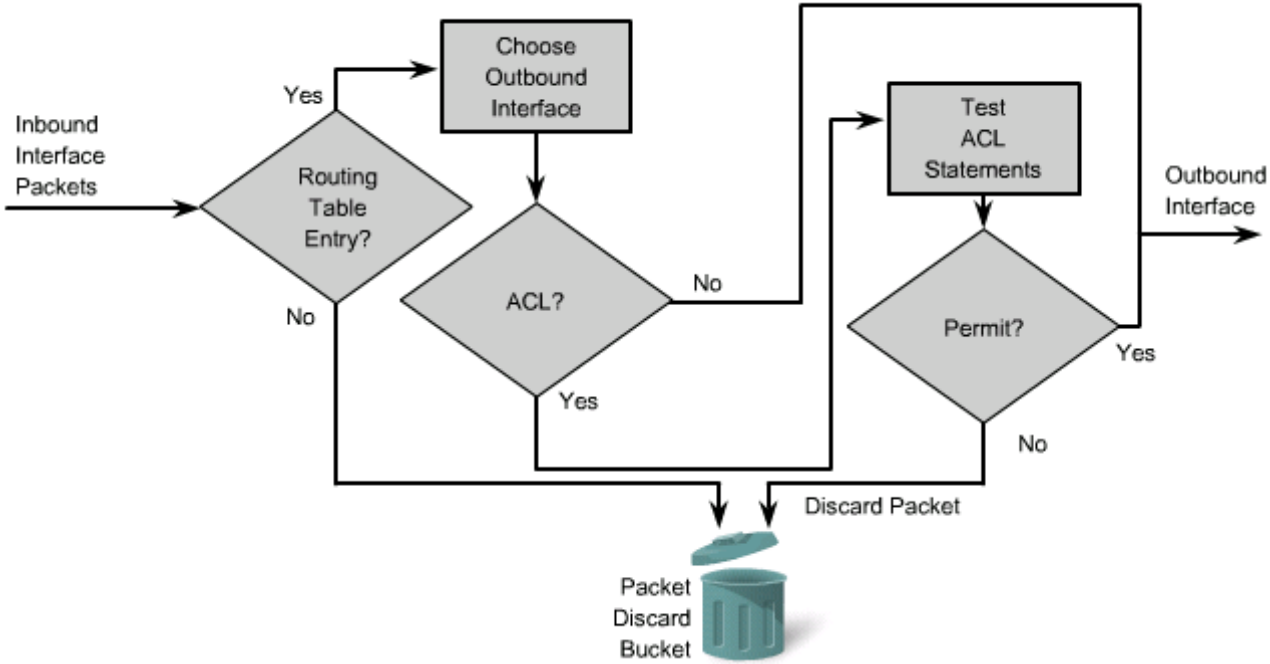
Как Cisco Routers обрабатывают правила ACL

Inbound ACL Operation Flow



Как Cisco Routers обрабатывают правила ACL

Outbound ACL Operation Flow



Расширенные ACL назначаются на интерфейсы так же, как и стандартные

Например:

```
R1(config)# interface f0/1
R1(config-if)# ip access-group 103 out
R1(config-if)# ip access-group 104 in
```

Создание именованного расширенного ACL

- **R1(config)# ip access-list extended SURFING**- объявляем именованный расширенный ACL для исходящего трафика
  R1(config-ext-nacl)# pernit tcp 192.168.10.0 0.0.0.255 any eq 80 - создаем правила
  R1(config-ext-nacl)# pernit tcp 192.168.10.0 0.0.0.255 any eq 443
  R1(config)# access-list extended BROWSING - объявляем именованный расширенный ACL для входящего трафика
  R1(config-ext-nacl)# pernit tcp any 192.168.10.0 0.0.0.255 established - создаем правила

Комплексные ACL

- Стандартные и расширенные ACL могут быть основой для комплексных ACL для улучшения функциональности.
- Бывают:
  - Динамические
  - С ограничением по времени
  - Established
  - Рефлексивные

Динамические ACL

- Если пользователю необходимо получить доступ к какому-либо устройству, находящемуся за маршрутизатором, он сначала должен аутентифицироваться на маршрутизаторе через Telnet. После этого маршрутизатор на определенное время дает доступ, по истечении которого опять надо будет аутентифицироваться.

- **R1(config)#username Student password 0 cisco** — создаем пользователей для подключения через Telnet без привелегий.
  R3(config)#access-list 101 permit tcp any host 10.2.2.2 eq telnet - разрешаем отовсюду подключаться к маршрутизатору
  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 - добавляем динамическую запись с именем testlist, которая будет работать только после установления связи через Telnet в течении 15 минут, а затем будет отключаться. Это правило открывает доступ из сети 192.168.10.0 в сеть 192.168.30.0.
  R3(config)#interface serial 0/0/1
  R3(config-if)#ip access-group 101 in — закрепляем ACL 101 за интерфейсом во 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 залогинится на маршрутизатор, выолнится автокоманда, которая даст доступ к сети 192.168.30.0. Сессия Telnet после этого закроется. Доступ к сети сохраниться и будет закрыт после 5 минут ожидания.

ACL с ограничением по времени

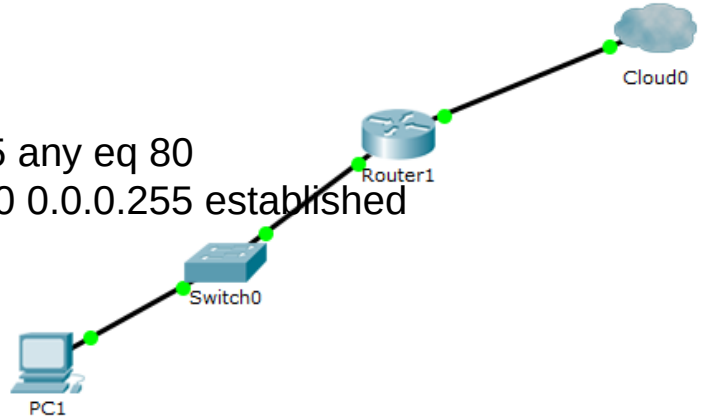
- Определяет время, когда в расширенном ACL работает конкретная запись
- **R1(config)#time-range EVERYOTHERDAY** - обьявляем переменную времени EVERYOTHERDAY.
  R1(config-time-range)#periodic Monday Wednesday Friday 8:00 to 17:00 — присваиваем список времени для этой переменной, в котором добавляем дни 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 переменную к правилу.
  R1(config)#interface s0/0/0
  R1(config-if)#ip access-group 101 out — закрепляем ACL за интерфейсом.

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.
- Следует отметить, что 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 ACL

Established в ACL Пример

- Настройка будет выглядеть так:
  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. Условно говоря, 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 в нашу внутреннюю сеть, и, самое главное только established трафик, то есть только трафик в рамках сессии которую установили мы изнутри.

Зеркальные (reflexive) ACL

- **Рефлексивные ACL** - разрешают трафик извне сети только в случае, если он был инициирован изнутри. Изначально весть трафик извне закрыт. Список доступа запоминает параметры пользовательских сессий, которые дают запрос наружу. Ответ на эти запросы проверяется на соответствие параметрам пользовательской сессии. Рефлексивные ACL имеют только временные записи, которые создаются автоматически с каждой сессией. Рефлексивные ACL не применяются непосредственно на интерфейс, но вкладываются в расширенный ACL , который применяется на интерфейс. Рефлексивные ACL могут быть определены только в расширенных именованных ACL, а использоваться могут с любими ACL.

- Технология эта напоминает внешне использование ключевого слова *established*, но имеется рад серьёзных отличий как в реализации, так и по функционалу. Суть технологии вот в чём: 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.

Зеркальные (reflexive) ACL Пример

- Пишется на выход следующие 2 ACL:

- R1(config)#ip access-list extended IN-TO-OUT
  R1(config-ext-nacl)#permit tcp 192.168.0.0 0.0.0.255 any eq www reflect BACK-WWW
  R1(config-ext-nacl)#permit tcp 192.168.0.0 0.0.0.255 any eq pop3 reflect BACK-POP
  R1(config-ext-nacl)#permit tcp 192.168.0.0 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#configure terminal
  Enter configuration commands, one per line. End with CNTL/Z.
  R1(config)#interface fa0/0
  R1(config-if)#ip access-group IN-TO-OUT in
  R1(config-if)#inter fa0/1
  R1(config-if)#ip access-group OUT-TO-IN in
  R1(config-if)#

Зеркальные (reflexive) ACL Пример

- 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 192.168.0.100 к веб страничке на сервере 123.123.123.123 при обращении выбирается случайный порт отправителя (например, 1234), 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 192.168.0.100 eq 1234