



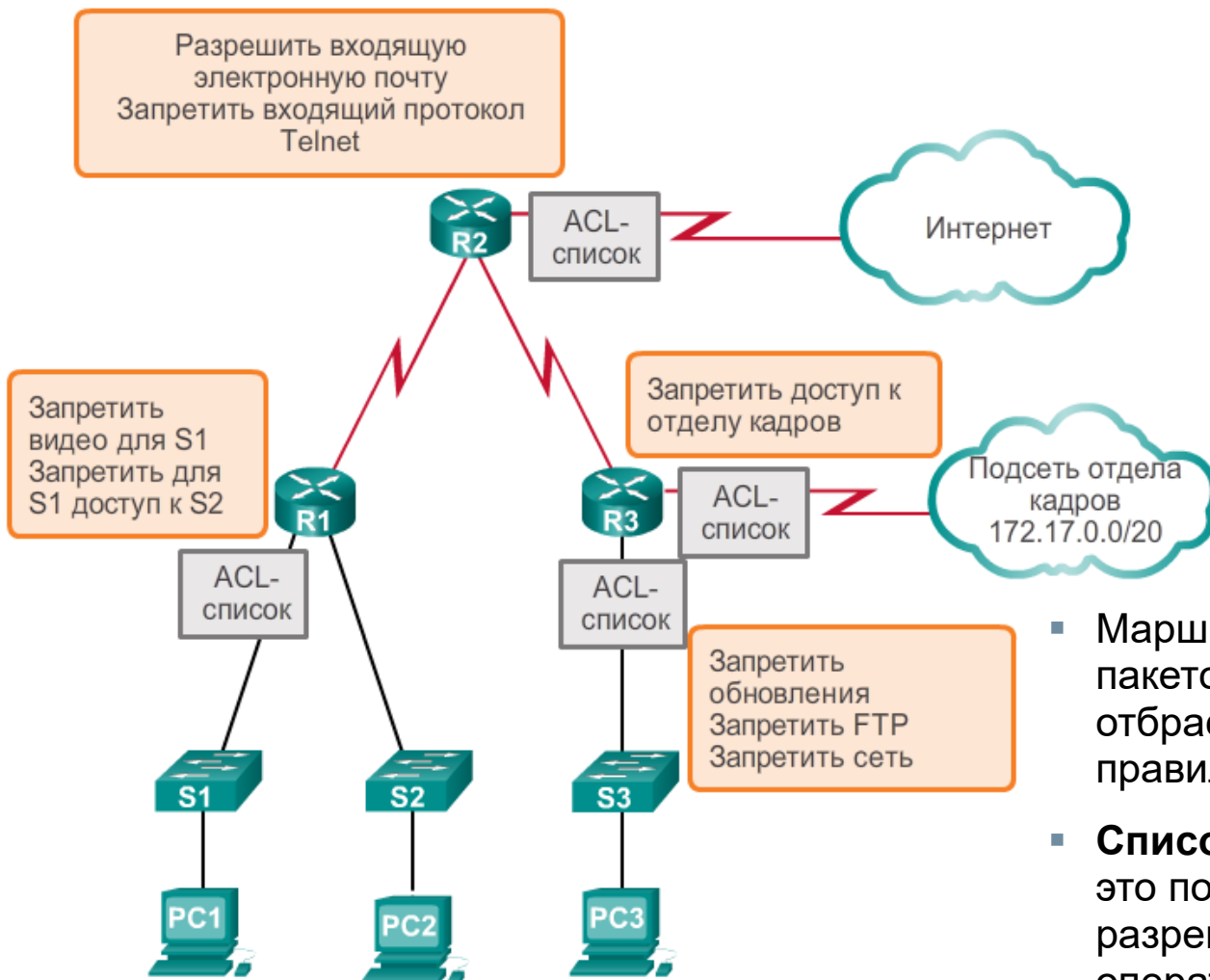
Списки контроля доступа (ACL)



**Корпоративные сети, безопасность и
автоматизация**

Назначение ACL-списков

Что такое список ACL?

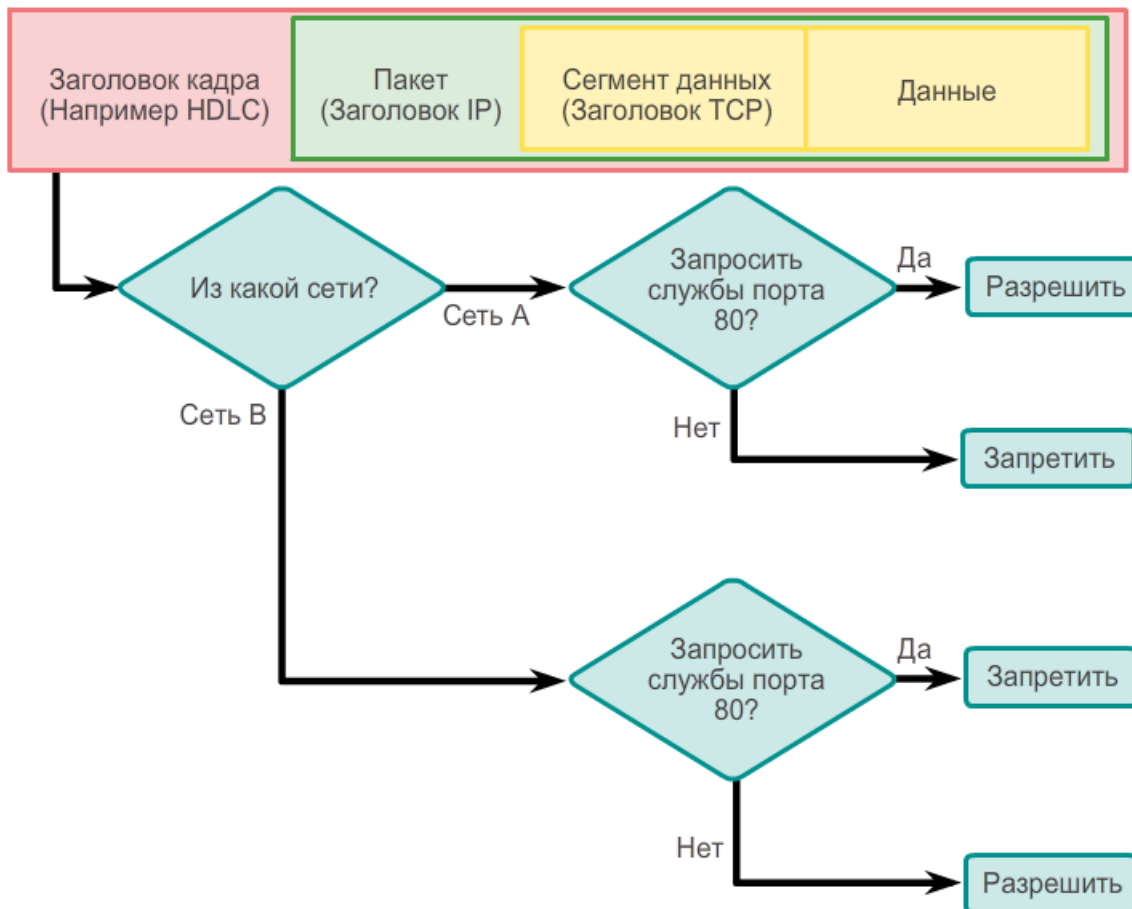


- С помощью фильтрации пакетов осуществляется управление доступом к сети путём анализа входящих и исходящих пакетов и пропуска или отбрасывания пакетов на основе заданных критериев, например, IP-адреса источника, IP-адреса назначения и протокола внутри пакета.
- Маршрутизатор работает как фильтр пакетов, когда перенаправляет или отбрасывает пакеты на основе правил фильтрации.
- Список контроля доступа ACL** — это последовательный список разрешающих или запрещающих операторов, называемых записями контроля доступа (ACE).

Назначение ACL-списков

Фильтрация пакетов

Пример фильтрации пакетов



Для оценки сетевого трафика, ACL-список извлекает следующую информацию из заголовка пакета уровня 3:

- IP-адрес источника;
- IP-адрес назначения;
- тип сообщения протокола ICMP.

ACL-список также может извлекать информацию более высокого уровня из заголовка уровня 4, включая:

- порт источника TCP/UDP;
- порт назначения TCP/UDP.

Назначение ACL-списков

Принцип работы ACL-списков



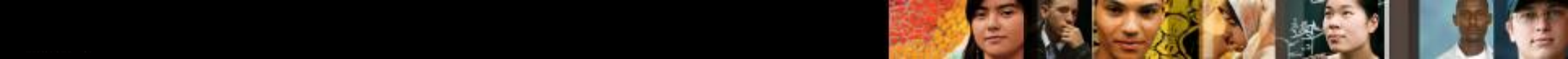
Входящий ACL-список фильтрует пакеты, приходящие на определённый интерфейс, до того, как они будут направлены на исходящий интерфейс.

Входящие ACL-списки являются оптимальным решением для фильтрации пакетов, когда сеть, подключенная к входящему интерфейсу, является единственным источником пакетов, требующих анализа.

Исходящий ACL-список фильтрует пакеты после их маршрутизации вне зависимости от входящего интерфейса.

Исходящие ACL-списки лучше всего использовать, когда одинаковые фильтры применяются к пакетам, поступающим с множества входящих интерфейсов, перед выходом на тот же исходящий интерфейс.

Последняя запись в ACL-списке всегда содержит **косвенный запрет трафика**. Данное правило автоматически вставляется в конец каждого ACL-списка, хотя и не присутствует в нём физически. Косвенный запрет блокирует весь трафик. Из-за косвенного запрета ACL-список, не содержащий хотя бы одного разрешающего правила, заблокирует весь трафик.



Сравнение стандартных и расширенных ACL-списков для IPv4

Типы ACL-списков для IPv4

Стандартные списки контроля доступа

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

Стандартные ACL-списки фильтруют IP-пакеты, исходя только из адреса источника.

Расширенные списки контроля доступа

```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

Расширенные ACL-списки фильтруют IP-пакеты, исходя из нескольких признаков, включая следующие:

- IP-адреса источника и назначения;
- порты TCP и UDP источника и назначения;
- тип протокола/номер протокола (например, IP, ICMP, UDP, TCP и т. д.).



Сравнение стандартных и расширенных ACL-списков для IPv4

Присваивание номеров и имён ACL-спискам

Нумерованный ACL-список:

Номер присваивается в зависимости от того, какой протокол будет фильтроваться.

- (От 1 до 99) и (от 1300 до 1999): стандартный ACL-список протокола IP
- (От 100 до 199) и (от 2000 до 2699): расширенный ACL-список протокола IP

Именованный ACL-список

Имя присваивается для определения ACL-списка.

- Имена могут содержать буквенно-цифровые символы.
- Рекомендуется вводить имя, используя ЗАГЛАВНЫЕ БУКВЫ.
- В именах не допускается наличие пробелов или знаков препинания.
- Записи ACL-списка можно добавлять или удалять.

Примерны ключевых слов

Пример 1:

```
R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255  
R1(config)# access-list 1 permit any
```

Пример 2:

```
R1(config)# access-list 1 permit 192.168.10.10 0.0.0.0  
R1(config)# access-list 1 permit host 192.168.10.10
```

Ключевое слово **permit** – разрешать

Ключевое слово **deny** - запрещать

Ключевое слово **host** применяется для маски 0.0.0.0. Эта маска указывает, что должны совпадать все биты IPv4-адреса, или совпадает только один узел.

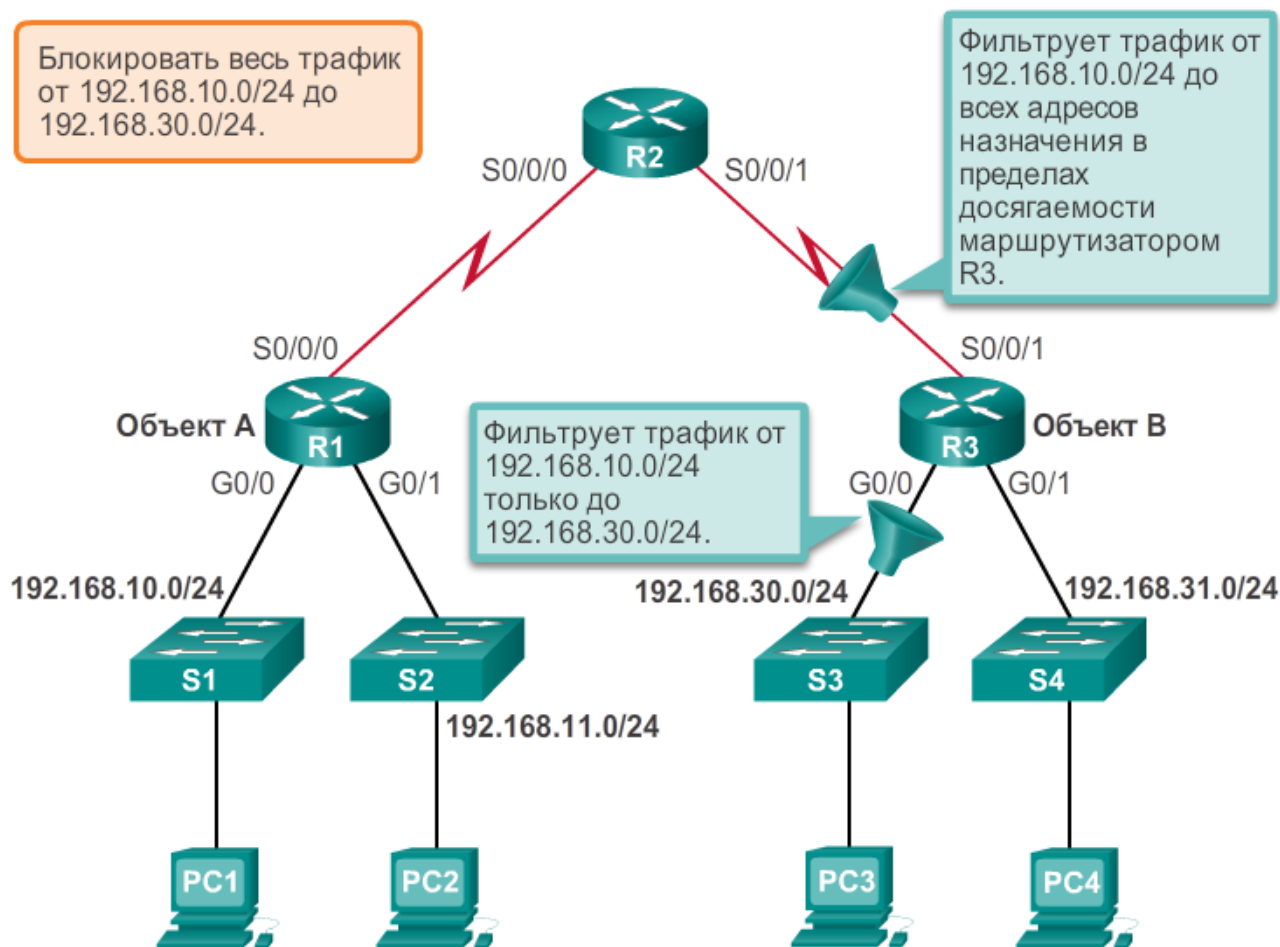
Ключевое слово **any** применяется для маски 255.255.255.255. Эта маска указывает, что необходимо игнорировать весь IPv4-адрес или принять любой адрес.

Общие рекомендации по созданию списков контроля доступа

Три «для»:

- **Один ACL-список для одного протокола.** Для управления потоком трафика на интерфейсе ACL-список должен быть определён для **каждого** протокола, действующего на интерфейсе.
- **Один ACL-список для одного направления.** ACL-списки способны одновременно контролировать трафик на одном направлении одного интерфейса. **Для управления исходящим и входящим трафиком должны быть созданы два отдельных ACL-списка.**
- **Один ACL-список для одного интерфейса.** ACL-списки управляют трафиком на одном интерфейсе.

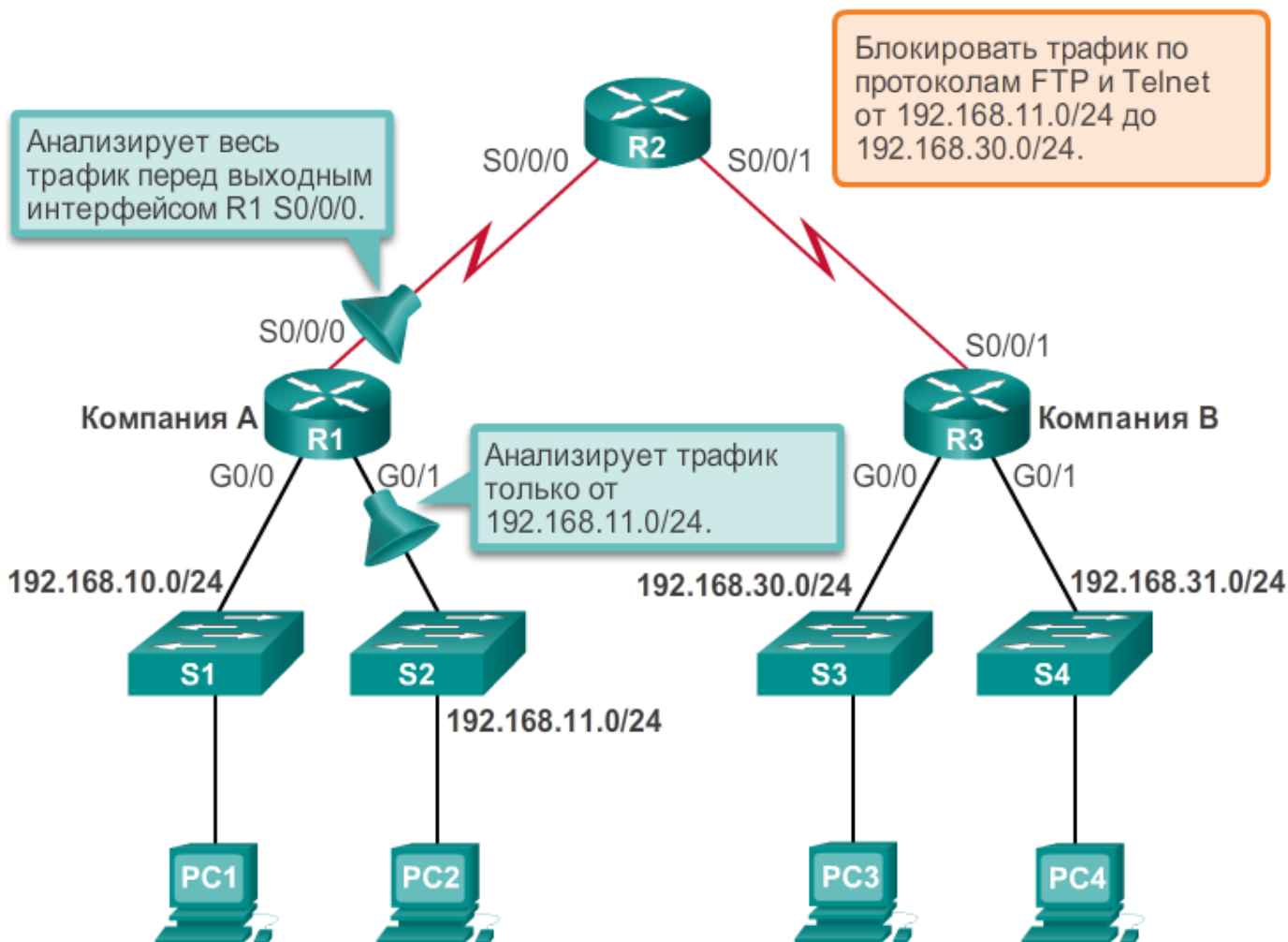
Размещение стандартного ACL-списка



Стандартные ACL-списки: поскольку стандартные списки контроля доступа не определяют адреса назначения, их размещают **максимально близко к месту назначения.**

Рекомендации по размещению списков ACL

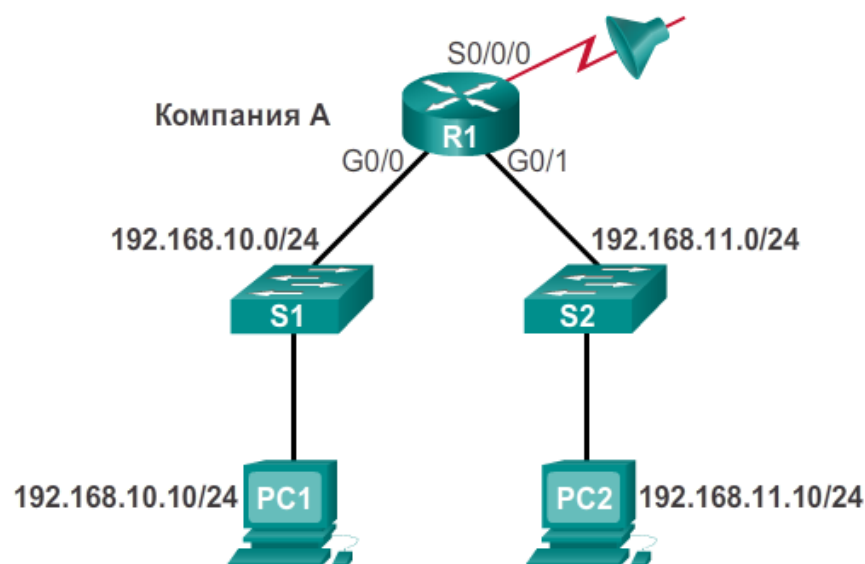
Размещение расширенного ACL-списка



Расширенные ACL-списки: расширенные ACL-списки следует размещать **максимально близко к источнику** фильтруемого трафика

Настройка стандартного ACL-списка IPv4

Последовательность ввода записей. Запрет трафика.



ACL-список 1

```
R1(config)#access-list 1 permit ip 192.168.10.0 0.0.0.255
```

ACL-список 2

```
R1(config)#access-list 2 permit ip 192.168.10.0 0.0.0.255  
R1(config)#access-list 2 deny any
```

Когда трафик поступает на маршрутизатор, он сравнивается с записями ACE в порядке, заданном в ACL-списке. Маршрутизатор продолжает обработку ACE, пока не обнаружит совпадение. Маршрутизатор обрабатывает пакет на основе первого найденного совпадения, остальные ACE-записи маршрутизатором не учитываются. Если к концу списка совпадения не найдены, маршрутизатор отклоняет трафик.

Применение ACL-списка 1 или ACL-списка 2 даёт одинаковые результаты. Для сети 192.168.10.0 будет разрешён доступ к сетям, достигаемым через S0/0/0.

Настройка стандартного ACL-списка

Полный синтаксис команды стандартного ACL-списка:

```
Router(config)# access-list номер-списка-доступа  
deny/permit/remark источник [ шаблонная маска-  
источника ] [ log ]
```

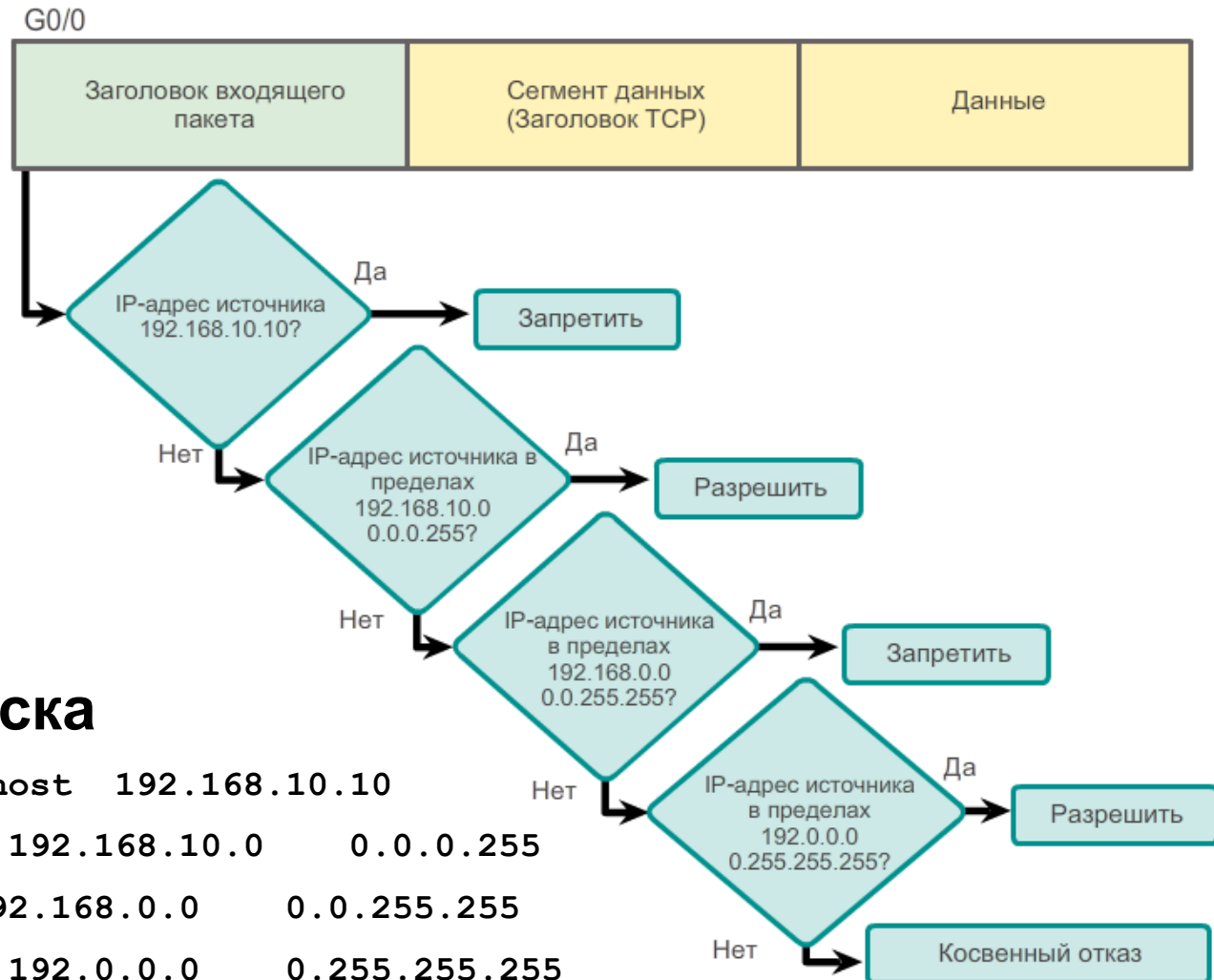
Для удаления ACL-списка применяется команда глобальной конфигурации **no access-list**.

Для документирования используется ключевое слово **remark**, которое также позволяет значительно облегчить восприятие списков контроля доступа.

Для ведения журнала по работе ACL-списков используется слово **log**.

Настройка стандартного ACL-списка IPv4

Настройка стандартного списка ACL



Пример ACL-списка

- `access-list 2 deny host 192.168.10.10`
- `access-list 2 permit 192.168.10.0 0.0.0.255`
- `access-list 2 deny 192.168.0.0 0.0.255.255`
- `access-list 2 permit 192.0.0.0 0.255.255.255`

Обработки записей стандартного списка доступа

Записи списка доступа обрабатываются последовательно. Поэтому важно соблюдать порядок их ввода.

```
R1(config)#access-list 3 deny 192.168.10.0 0.0.0.255  
R1(config)#access-list 3 permit host 192.168.10.10  
% Access rule can't be configured at higher sequence num as  
it is part of the existing rule at sequence num 10  
R1(config)#
```

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

Применение стандартных ACL-списков на интерфейсах

После создания стандартного ACL-списка, его назначают интерфейсу с помощью команды режима настройки интерфейса **ip access-group**:

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

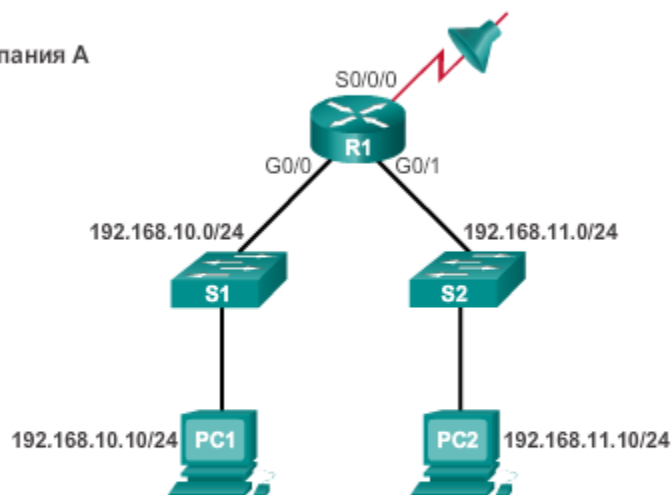
Для удаления ACL-списка из интерфейса сначала следует ввести команду **no ip access-group** на интерфейсе, а затем — глобальную команду **no access-list** для удаления всего ACL-списка.

Настройка стандартного ACL-списка IPv4

Применение стандартных ACL-списков на интерфейсах

Разрешить определенную подсеть

Компания А



```
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
```

Шаг 1: с помощью команды глобальной конфигурации **access-list** создайте запись в стандартном ACL-списке IPv4.

```
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
```

Запись в примере совпадает с любым адресом, который начинается с 192.168.10.x. Используйте параметр **remark**, чтобы добавить описание к списку контроля доступа.

Шаг 2: используйте команду конфигурации **interface**, чтобы выбрать интерфейс, на котором следует применить ACL-список.

```
R1(config)# interface serial 0/0/0
```

Шаг 3: используйте команду конфигурации интерфейса **ip access-group**, чтобы активировать существующий ACL-список на интерфейсе.

```
R1(config-if)# ip access-group 1 out
```

В этом примере стандартный список IPv4 ACL 1 активируется на интерфейсе в качестве исходящего фильтра.

Создание именованного стандартного ACL-списка

```
Router(config)# ip access-list [standard | extended] name
```

Строка с буквенно-цифровым именем должна быть уникальной и не должна начинаться с цифры.

```
Router(config-std-nacl)# [permit | deny | remark] {source  
[source-wildcard]} [log]
```

```
Router(config-if)# ip access-group name [in | out]
```

Активирует на интерфейсе именованный ACL-список по протоколу IP.

Настройка стандартного ACL-списка IPv4

Комментарии к ACL-спискам

Пример 1. Комментарии к нумерованному ACL-списку

```
R1(config)# access-list 1 remark Do not allow Guest workstation through
R1(config)# access-list 1 deny host 192.168.10.10
R1(config)# access-list 1 remark Allow devices from all other 192.168.x.x subnets
R1(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 1 out
R1(config-if)#
```

Пример 2. Комментарии к именованному ACL-списку

```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# remark Do not allow access from Lab workstation
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# remark Allow access from all other networks
R1(config-std-nacl)# permit any
R1(config-std-nacl)# exit
R1(config-std-nacl)# interface G0/0
R1(config-if)# ip access-group NO_ACCESS out
R1(config-if)#
```

Редактирование стандартного нумерованного ACL-списка

Конфигурация

```
R1(config)# access-list 1 deny host 192.168.10.99  
R1(config)# access-list 1 permit 192.168.0.0 0.0.255.255
```

Шаг 1

```
R1# show running-config | include access-list 1  
access-list 1 deny host 192.168.10.99  
access-list 1 permit 192.168.0.0 0.0.255.255
```

Шаг 2

```
R1# config t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)# no access-list 1  
R1(config)# access-list 1 deny host 192.168.10.10  
R1(config)# access-list 1 permit 192.168.0.0 0.0.255.255
```

Шаг 3

```
R1# show running-config | include access-list 1  
access-list 1 deny host 192.168.10.10  
access-list 1 permit 192.168.0.0 0.0.255.255
```

Редактирование стандартного нумерованного ACL-списка (продолжение)

Редактирование нумерованных ACL-списков с помощью порядковых номеров

Конфигурация

```
R1(config)# access-list 1 deny host 192.168.10.99
R1(config)# access-list 1 permit 192.168.0.0 0.0.255.255
```

Шаг 1

```
R1# show access-lists 1
Standard IP access list 1
 10 deny 192.168.10.99
 20 permit 192.168.0.0, wildcard bits 0.0.255.255
R1#
```

Шаг 2

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1#
```

Шаг 3

```
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
R1#
```

Отобразите текущий ACL-список с помощью команды **show access-lists 1**. Порядковый номер отображается в начале каждой записи. Порядковый номер автоматически присваивается при добавлении записи в список.

Введите команду **ip access-lists standard**. Номер ACL-списка 1 используется как его имя. Сначала необходимо удалить некорректно сконфигурированную запись с помощью команды **no 10**, где 10 ссылается на порядковый номер. Затем добавьте новую запись с порядковым номером 10 при помощи команды **10 deny host 192.168.10.10**.

Редактирование стандартного именованного ACL-списка

Добавление строки в именованный ACL-список

```
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#
```

Примечание. Команда именованного ACL-списка `no sequence-number` применяется для удаления отдельных записей.

Преобразование ACL-списков IPv4

Проверка ACL-списка

```
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#
```


Преобразование ACL-списков IPv4

Статистика ACL-списка

```
R1# show access-lists
Standard IP access list 1
 10 deny 192.168.10.10 (4 match(es))
 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 (4 match(es))
 20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

Ответ PC3 на запрос от PC1.

```
R1# show access-lists
Standard IP access list 1
 10 deny 192.168.10.10 (8 match(es))
 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 (4 match(es))
 20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

Количество
совпадений
увеличилось.

Расширенные списки контроля доступа

Применение номеров портов

```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 23
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 20
```

Применение ключевых слов

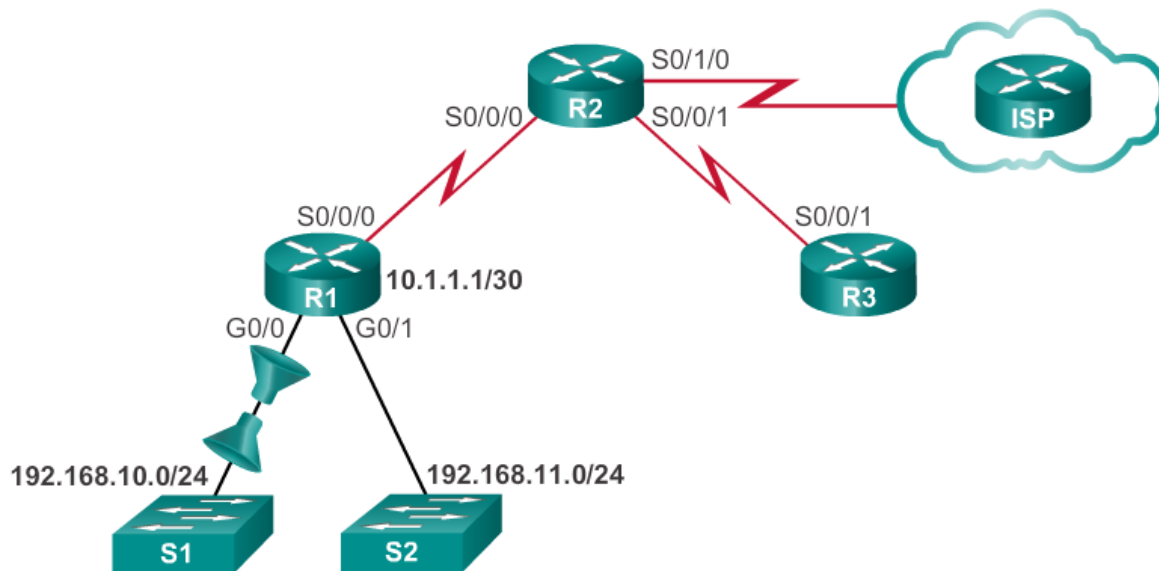
```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq telnet
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp-data
```

Возможные фильтры расширенных ACL-списков:

- Адрес источника
- Адрес назначения
- Протокол
- Номера портов

```
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]
```

Применение расширенных ACL-списков на интерфейсах



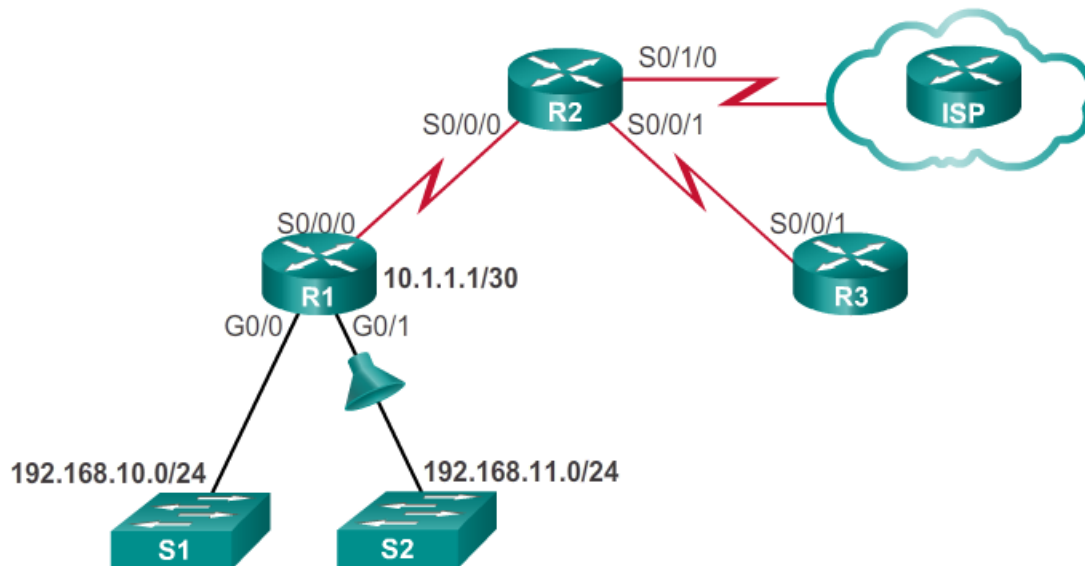
```
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)#access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)#interface g0/0
R1(config-if)#ip access-group 103 in
R1(config-if)#ip access-group 104 out
```

ACL 103 разрешает трафик, поступающий от любого адреса в сети 192.168.10.0, к любому месту назначения с учетом, что трафик использует только порты 80 (HTTP) и 443 (HTTPS). Характер протокола HTTP требует, чтобы трафик возвращался обратно в сеть от веб-сайтов, к которым обращались внутренние клиенты.

Параметр **established** разрешает возврат в сеть 192.168.10.0/24 только того трафика, который изначально исходил из этой сети.

Фильтрация трафика с использованием расширенных ACL-списков

Расширенный ACL-список для запрета протокола FTP



```
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255
192.168.10.0 0.0.0.255 eq ftp
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255
192.168.10.0 0.0.0.255 eq ftp-data
R1(config)# access-list 101 permit ip any any
R1(config)# interface g0/1
R1(config-if)# ip access-group 101 in
```

Запрещён трафик FTP из подсети 192.168.11.0, который направляется в подсеть 192.168.10.0, но разрешён любой другой тип трафика. Протокол FTP использует порты TCP 20 и 21; таким образом, для запрета доступа FTP ACL-списку требуется оба ключевых слова имени порта: **ftp** и **ftp-data** или **eq 20** и **eq 21**.

Проверка расширенных ACL-списков

```
R1#show access-lists
Extended IP access list BROWSING
  10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
  10 permit tcp 192.168.10.0 0.0.0.255 any eq www
  20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
R1#show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
<output omitted for brevity>
  Outgoing access list is BROWSING
  Inbound access list is SURFING
<output omitted for brevity>
```

Процесс принятия решений расширенного ACL-списка

Расширенный ACL-список выполняет фильтрацию:

1. по адресу источника,
2. по порту и протоколу источника,
3. по адресу назначения,
4. порту и протоколу назначения.

После этого принимается окончательное решение о разрешении или запрете.

Типы ACL-списков для IPv6



ACL-списки для IPv4

- Стандартный
 - Нумерованный
 - Именованный
- Расширенный
 - Нумерованный
 - Именованный

ACL-списки для IPv6

- Только именованный
- По функциональности аналогичен расширенному ACL-списку для IPv4



Создание ACL-списка IPv6

Сравнение ACL-списков для IPv4 и IPv6

Несмотря на то, что ACL-списки для IPv4 и IPv6 очень схожи, у них есть два серьёзных отличия.

- Применение ACL-списка для IPv6

IPv6 использует команду `ipv6 traffic-filter` для выполнения аналогичной функции на интерфейсах IPv6.

- Отсутствие шаблонных масок

Длина префикса используется для указания того, какая часть IPv6-адреса источника или назначения должна совпадать.

Настройка ACL-списков IPv6

Для настройки ACL-списка IPv6 нужно выполнить три основных действия:

- В режиме глобальной конфигурации используйте команду **ipv6 access-list** *укажите имя* для создания ACL-списка IPv6.
- Из режима конфигурации именованного ACL-списка примените разрешающую команду **permit** или запрещающую команду **deny** для указания одного или более условий, согласно которым пакет будет отправлен или отклонён.
- С помощью команды **end** вернитесь в привилегированный режим EXEC.

```
R1(config)# ipv6 access-list access-list-name  
R1(config-ipv6-acl)# deny | permit protocol {source-ipv6-  
prefix/prefix-length | any | host source-ipv6-address} [operator  
[port-number]] {destination-ipv6-prefix/ prefix-length | any |  
host destination-ipv6-address} [operator [port-number]]
```

Примеры ACL-списков для IPv6

Запрет протокола FTP

```
R1(config)#ipv6 access-list NO-FTP-TO-11
R1(config-ipv6-acl)#deny tcp any 2001:db8:cafe:11::/64 eq ftp
R1(config-ipv6-acl)#deny tcp any 2001:db8:cafe:11::/64 eq ftp-data
R1(config-ipv6-acl)#permit ipv6 any any
R1(config-ipv6-acl)#exit
R1(config)#interface g0/0
R1(config-if)#ipv6 traffic-filter NO-FTP-TO-11 in
R1(config-if)#
```

Настройка ACL-списков IPv6

Проверка ACL-списков IPv6

```
R3#show ipv6 interface g0/0
```

```
GigabitEthernet0/0 is up, line protocol is up
```

```
Global unicast address(es):
```

```
2001:DB8:CAFE:30::1, subnet is 2001:DB8:CAFE:30::/64
```

```
Input features: Access List
```

```
Inbound access list RESTRICTED-ACCESS
```

```
<some output omitted for brevity>
```

```
R3#show access-lists
```

```
IPv6 access list RESTRICTED-ACCESS
```

```
permit tcp any host 2001:DB8:CAFE:10::10 eq www sequence 20
```

```
permit tcp any host 2001:DB8:CAFE:10::10 eq 443 sequence 30
```

```
deny ipv6 any 2001:DB8:CAFE:10::/64 sequence 50
```

```
permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11 eq  
telnet sequence 70
```

```
deny tcp any host 2001:DB8:CAFE:11::11 eq telnet sequence 90
```

```
permit ipv6 any any sequence 110
```

```
R3#
```



Защита портов VTY с помощью стандартного списка контроля доступа IPv4

Команда `access-class`

Стандартный ACL-список может обеспечить удаленный административный доступ к устройству с помощью линий `vtu`, для этого необходимо:

- Создать список ACL, чтобы определить, каким административным узлам должен быть разрешен удаленный доступ.
- Применить ACL к входящему трафику на линиях `vtu`.

```
R1(config-line)# access-class {access-list-number | access-list-name} { in | out }
```

Защита портов VTY с помощью стандартного списка контроля доступа для IPv4

Пример безопасного доступа VTY

В этом примере показано, как настроить ACL для фильтрации трафика vty.

- Сначала настраивается запись локальной базы данных для пользовательского имени **ADMIN** с паролем **class**.
- Строки vty на R1 настроены на использование локальной базы данных для проверки подлинности, разрешение трафика SSH и использование ADMIN-HOST ACL для ограничения трафика.

```
R1(config)# username ADMIN secret class
R1(config)# ip access-list standard ADMIN-HOST
R1(config-std-nacl)# remark This ACL secures incoming vty lines
R1(config-std-nacl)# permit 192.168.10.10
R1(config-std-nacl)# deny any
R1(config-std-nacl)# exit
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input telnet
R1(config-line)# access-class ADMIN-HOST in
R1(config-line)# end
R1#
```



Защита портов VTY с помощью стандартного списка контроля доступа IPv4

Проверка безопасности порта VTY

После настройки ACL-списка для ограничения доступа к линиям VTY важно убедиться в его надлежащем функционировании.

Чтобы проверить статистику ACL, выполните команду **show access-lists** .

- Совпадение в строке разрешения выходных данных является результатом успешного SSH-соединения хоста с IP-адресом 192.168.10.10.
- Соответствие в операторе deny связано с неудачной попыткой создать соединение SSH с устройства в другой сети.

```
R1#  
Oct  9 15:11:19.544: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 192.168.10.10]  
[localport: 23] at 15:11:19 UTC Wed Oct 9 2019  
R1# show access-lists  
Standard IP access list ADMIN-HOST  
 10 permit 192.168.10.10 (2 matches)  
 20 deny    any (2 matches)  
R1#
```




Преобразование сетевых адресов IPv4



**Корпоративные сети, безопасность и
автоматизация**

Характеристики NAT

Терминология NAT

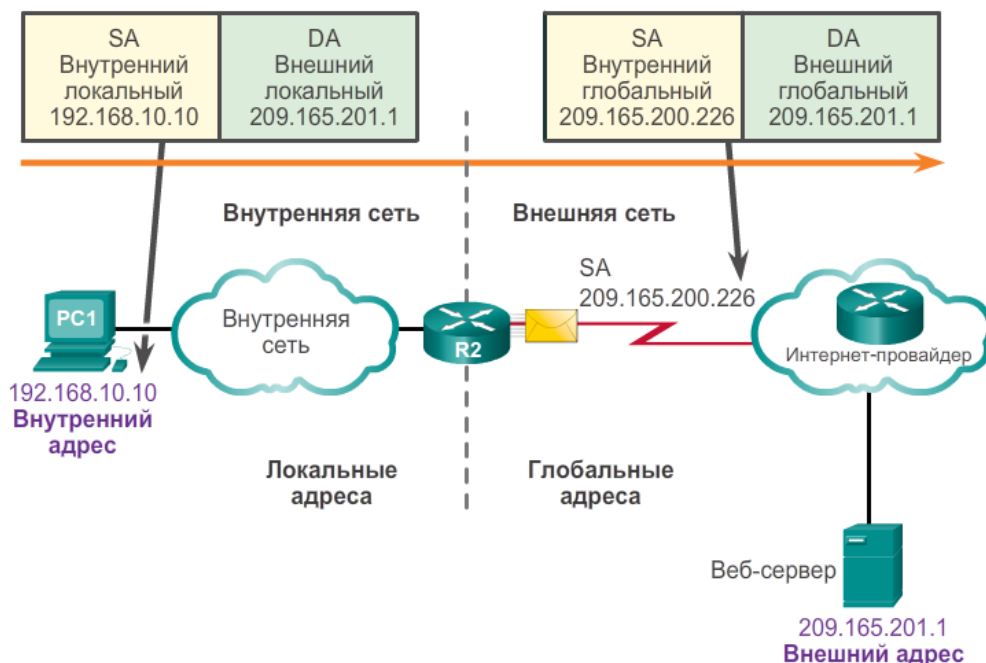
- Согласно терминологии NAT, внутренняя сеть представляет собой комплект устройств, использующих частные адреса. Внешними сетями являются все остальные сети.



- В NAT предусмотрено четыре

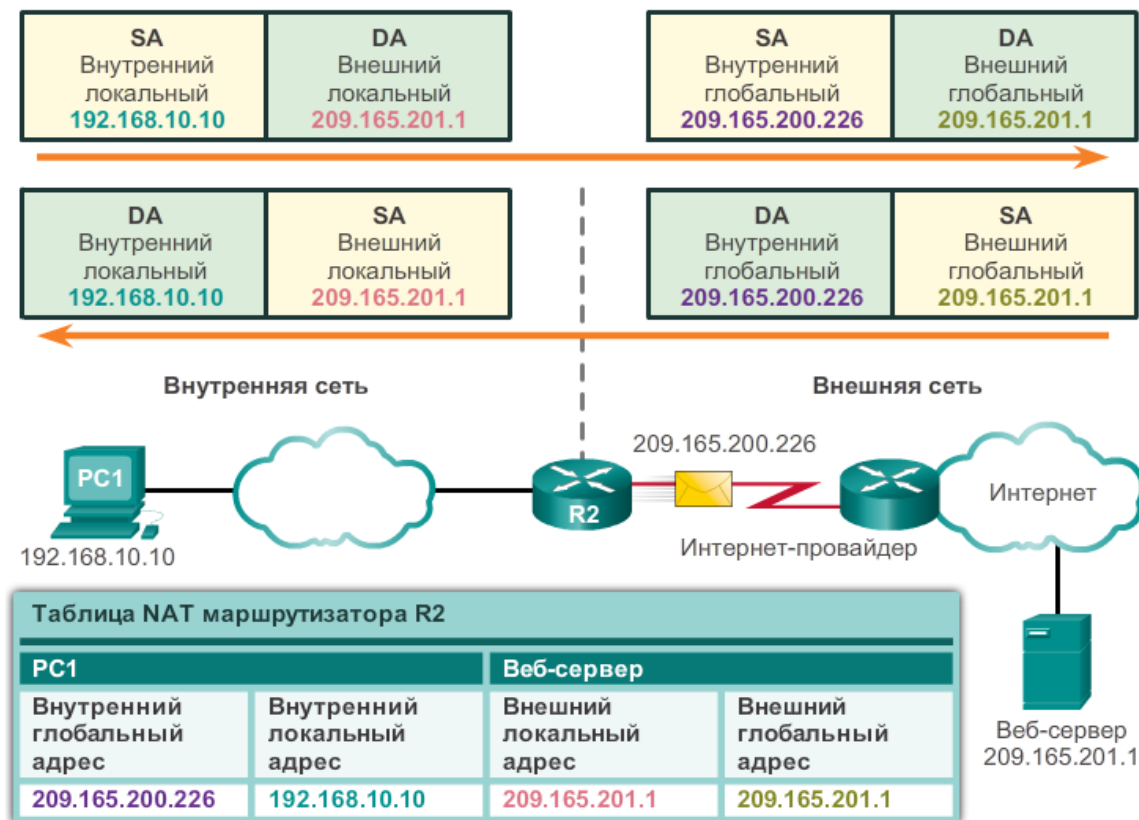
типа адресов:

- ✓ Внутренний локальный адрес;
- ✓ Внутренний глобальный адрес;
- ✓ Внешний локальный адрес;
- ✓ Внешний глобальный адрес.



Характеристики NAT

Принципы работы NAT



Внутренний локальный адрес — это *адрес источника, видимый из внутренней сети*. PC1 назначен IPv4-адрес 192.168.10.10. Это внутренний локальный адрес PC1.

Внутренний глобальный адрес — это *адрес источника, видимый из внешней сети*. Когда PC1 отправляет трафик веб-серверу с адресом 209.165.201.1, R2 преобразует внутренний локальный адрес во внутренний глобальный адрес. В этом случае R2 меняет исходный IPv4-адрес с 192.168.10.10 на 209.165.200.226. В терминологии NAT, внутренний локальный адрес 192.168.10.10 преобразуется во внутренний глобальный адрес 209.165.200.226.

Внешний глобальный адрес — это *адрес назначения, видимый из внешней сети*. Это глобально маршрутизируемый IPv4-адрес, назначенный узлу в Интернете. Например, веб-сервер доступен по IPv4-адресу 209.165.201.1. В большинстве случаев внешний локальный и внешний глобальный адреса совпадают.

Внешний локальный адрес — это *адрес назначения, видимый из внутренней сети*. В этом примере PC1 отправляет трафик веб-серверу с IPv4-адресом 209.165.201.1. В редких случаях этот адрес может отличаться от глобально маршрутизируемого адреса назначения.



Принцип работы NAT

Типы NAT

Существуют три механизма преобразования сетевых адресов:

- **Статическое преобразование сетевых адресов (статический NAT)** — это взаимно-однозначное соответствие между локальным и глобальным адресами.
- **Динамическое преобразование сетевых адресов (динамический NAT)** — это сопоставление адресов по схеме «многие ко многим» между локальными и глобальными адресами.
- **Преобразование адресов портов (PAT)** — это сопоставление адресов по схеме «многие к одному» между локальными и глобальными адресами. Данный метод также называется перегрузкой (NAT с перегрузкой).

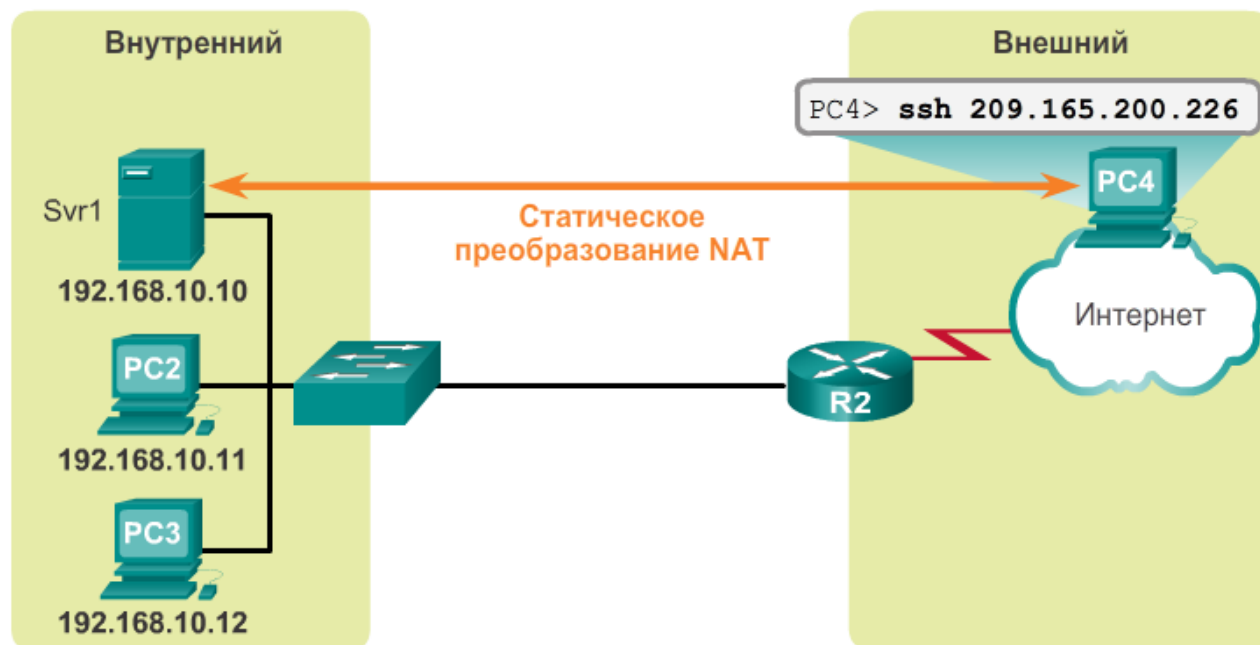
Типы NAT

Статический NAT

Статическое преобразование сетевых адресов (NAT)

Таблица статического NAT

Внутренний локальный адрес	Внутренний глобальный адрес — адреса доступны через маршрутизатор R2
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228



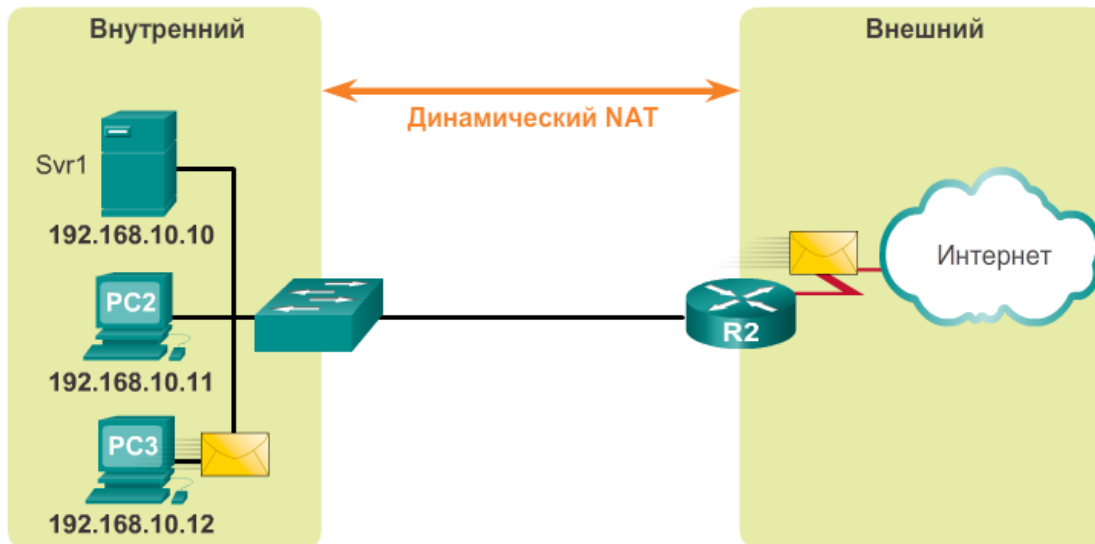
- Статический NAT использует сопоставление локальных и глобальных адресов по схеме «**ОДИН-В-ОДИН**».
- Эти соответствия задаются администратором сети и остаются **неизменными**.
- Статический метод преобразования NAT особенно полезен, **если серверы внутренней сети должны быть доступны из внешней сети**.
- Сетевой администратор может получить доступ по протоколу SSH к серверу, расположенному во внутренней сети, указав своему SSH-клиенту соответствующий внутренний глобальный адрес.

Типы NAT

Динамический NAT

Динамическое преобразование сетевых адресов NAT

IPv4 NAT-пул	
Внутренний локальный адрес	Внутренний пул глобальных адресов — адреса доступны через маршрутизатор R2
192.168.10.12	209.165.200.226
Доступен	209.165.200.227
Доступен	209.165.200.228
Доступен	209.165.200.229
Доступен	209.165.200.230



- Метод динамического преобразования сетевых адресов (динамический NAT) использует **пул публичных адресов**, которые присваиваются **в порядке живой очереди**.
- Когда внутреннее устройство запрашивает доступ к внешней сети, **динамический NAT присваивает доступный публичный IPv4-адрес из пула**.
- Один внутренний адрес преобразуется в один внешний адрес.
- Для динамического NAT требуется **достаточное количество публичных адресов, доступных для общего количества одновременных сеансов пользователей**.

Типы NAT

Преобразование порт-адрес NAT (PAT)

Процесс PAT

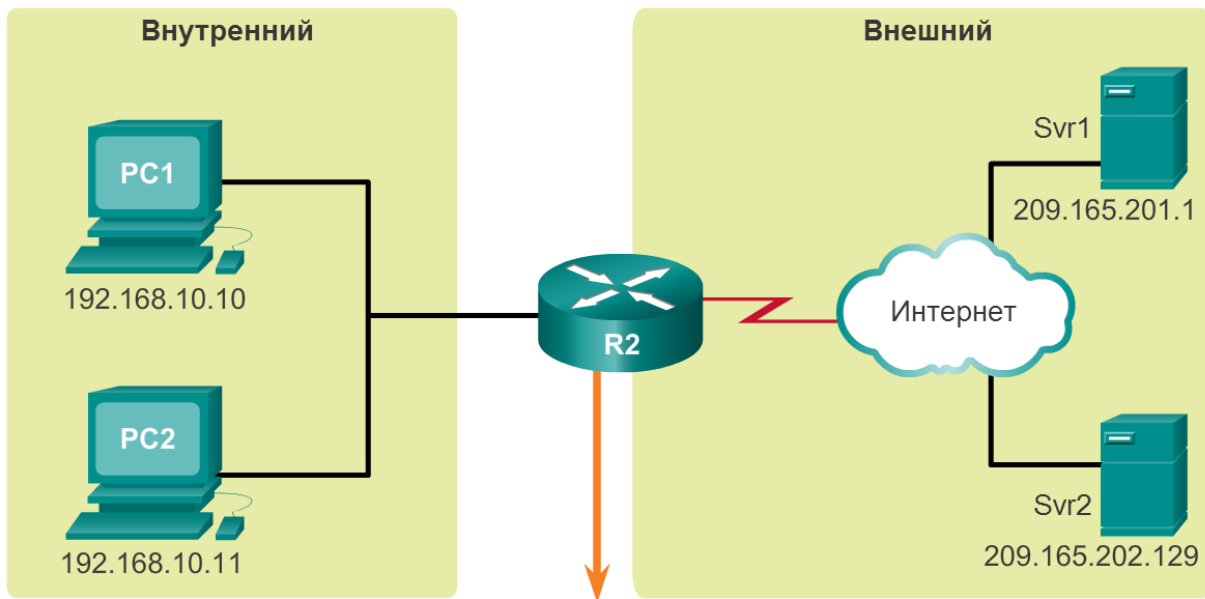


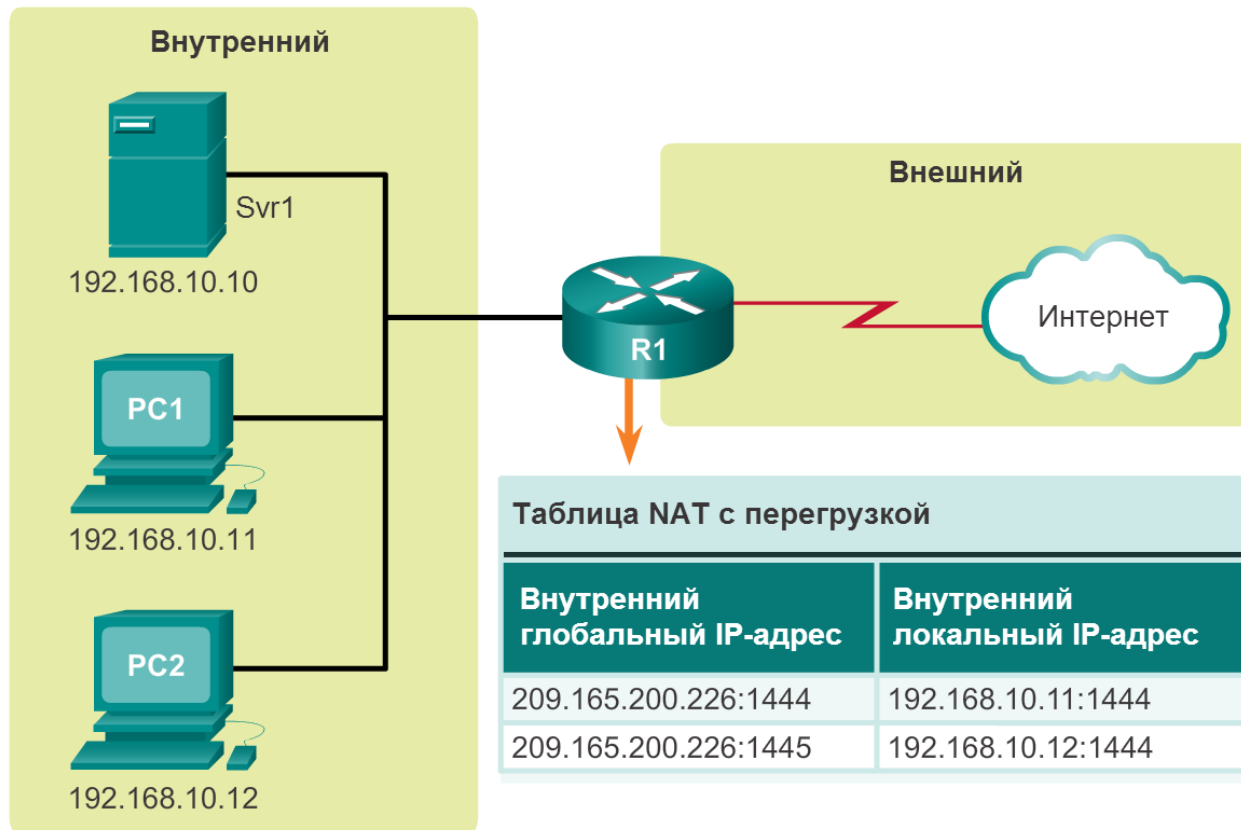
Таблица NAT с перегрузкой

Внутренний глобальный IP-адрес	Внутренний локальный IP-адрес	Внешний локальный IP-адрес	Внешний глобальный IP-адрес
209.165.200.226:1555	192.168.10.10:1555	209.165.201.1:80	209.165.201.1:80
209.165.200.226:1331	192.168.10.11:1331	209.165.202.129:80	209.165.202.129:80

- PAT сопоставляет **множество частных IPv4-адресов** одному или нескольким публичным IPv4-адресам.
- Используя пару **порт-источник** и **IP-адрес источника**, PAT отслеживает от какого внутреннего клиента идёт тот или иной трафик
- PAT известен также под названием «NAT с перегрузкой».
- Используя номер порта, с помощью PAT можно отправлять пакеты ответа верному внутреннему устройству.
- При помощи процесса PAT также осуществляется подтверждение, что входящие пакеты действительно были запрошены. Таким образом, повышается степень безопасности сеанса.

Типы NAT

Преобразование порт-адрес NAT (PAT)



Узлы выбирают **один и тот же номер порта — 1444.**

Это допустимо для внутреннего адреса, поскольку узлам назначены уникальные частные IP-адреса. Но на маршрутизаторе с поддержкой NAT номера портов необходимо изменить. В противном случае пакеты от двух различных узлов выходили бы из R1 с одинаковым адресом источника. В

рассматриваемом примере в процессе преобразования PAT второму адресу узла назначается следующий доступный порт (1445).

PAT	
Внутренний глобальный адрес	Внутренний локальный адрес
209.165.200.226:1444	192.168.10.10:1444
209.165.200.226:1445	192.168.10.11:1444
209.165.200.226:1555	192.168.10.12:1555
209.165.200.226:1556	192.168.10.13:1555



Типы NAT

Сравнение NAT и PAT

NAT - изменяет только адреса IPv4

Внутренний глобальный адрес	Внутренний локальный адрес
209.165.200.226	192.168.10.10

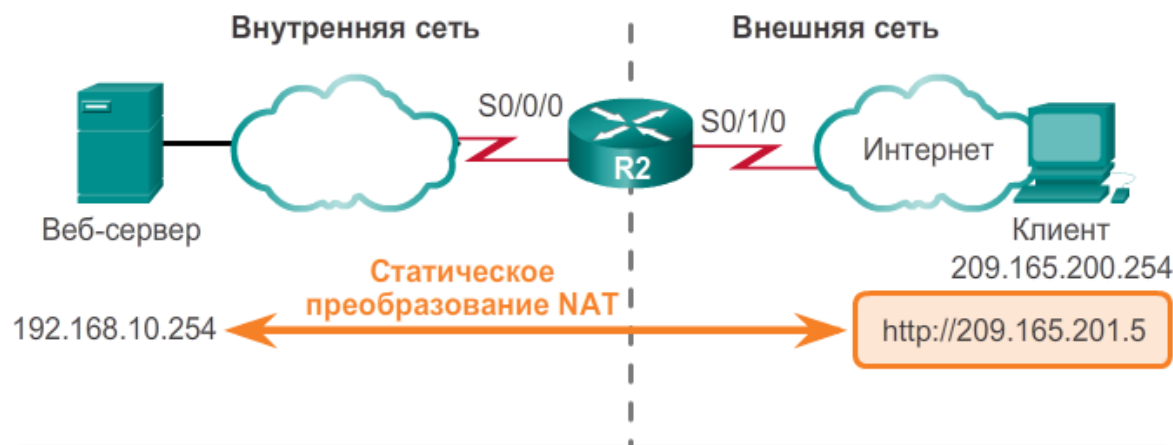
PAT - меняет и адрес, и номер порта

Внутренний глобальный адрес	Внутренний локальный адрес
209.165.200.226:2031	192.168.10.10:2031

NAT	PAT
Сопоставление локальных и глобальных адресов по схеме «один к одному»	Один внутренний глобальный адрес может быть сопоставлен со многими внутренними локальными адресами.
В процессе преобразования использует только адреса IPv4.	Использует IPv4 адреса и номера портов источника TCP или UDP в процессе преобразования.
Уникальный внутренний глобальный адрес необходим для каждого внутреннего узла, обращающегося к внешней сети.	Один уникальный внутренний глобальный адрес может быть общим для многих внутренних узлов, обращающихся к внешней сети.

Настройка статического NAT

Пример конфигурации статического NAT



```
Establishes static translation between an inside local address and
an inside global address.
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5

R2(config)# interface Serial0/0/0
R2(config-if)# ip address 10.1.1.2 255.255.255.252
Identifies interface serial 0/0/0 as an inside NAT interface.
R2(config-if)# ip nat inside
R2(config-if)# exit

R2(config)# interface Serial0/1/0
R2(config-if)# ip address 209.165.200.225 255.255.255.224
Identifies interface serial 0/1/0 as the outside NAT interface.
R2(config-if)# ip nat outside
```

Настройка статического NAT сопряжена с двумя основными задачами:

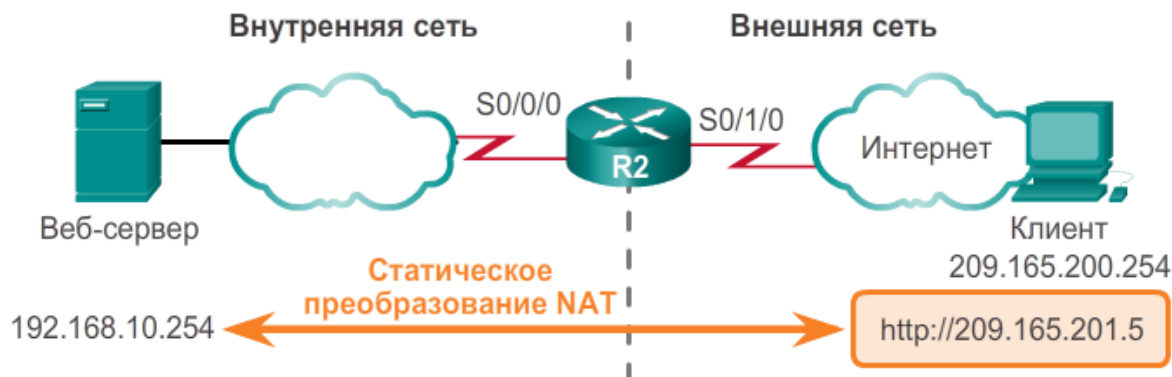
- Созданием соответствия между внутренним локальным и внешним локальным адресами и
- Определением двух интерфейсов - принадлежащих внутренней сети и внешней сети.

Создание соответствия между внутренним локальным 192.168.10.254 и внутренним глобальным 209.165.201.5 адресами.

После настройки соответствия интерфейсы, участвующие в преобразовании, настраиваются как внутренние или внешние относительно NAT. В этом примере интерфейс **Serial 0/0/0** маршрутизатора R2 является **внутренним**, а **Serial 0/1/0** — **внешним** интерфейсом.

Настройка статического NAT

Проверка статического NAT



Статическое преобразование всегда представлено в таблице NAT.

```
R2# show ip nat translations
Pro Inside global  Inside local  Outside local  Outside global
--- 209.165.201.5  192.168.10.254  ---          ---
R2#
```

Статическое преобразование во время активного сеанса.

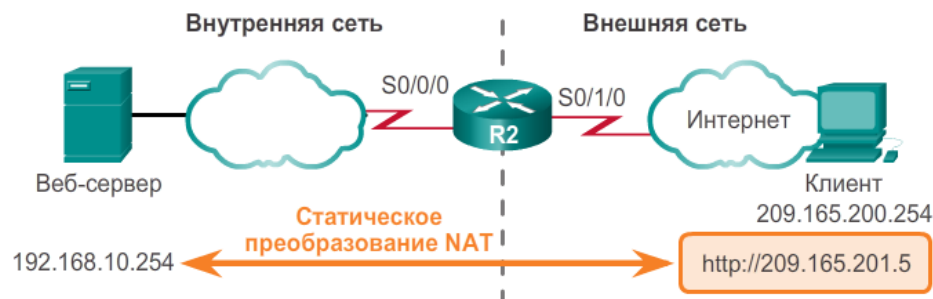
```
R2# show ip nat translations
Pro Inside global  Inside local  Outside local  Outside global
--- 209.165.201.5  192.168.10.254  209.165.200.254  209.165.200.254
R2#
```

Команда **show ip nat translations** отображает активные преобразования NAT. Статические преобразования всегда присутствуют в таблице NAT независимо от активных взаимодействий.

Если команда вводится во время активного сеанса, выходные данные будут также содержать адрес внешнего устройства.

Настройка статического NAT

Проверка статического NAT



```
R2# clear ip nat statistics

R2# show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Peak translations: 0
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  Serial0/0/0
Hits: 0 Misses: 0
<выходные данные опущены>

Client PC establishes a session with the web server

R2# show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Peak translations: 2, occurred 00:00:14 ago
Outside interfaces:
  Serial0/1/0
Inside interfaces:
  Serial0/0/0
Hits: 5 Misses: 0
<выходные данные опущены>
```

Команда **show ip nat statistics** выводит сведения о суммарном количестве активных преобразований.

До начала взаимодействия с веб-сервером команда **show ip nat statistics** не должна показывать каких-либо совпадений. После установки клиентом сеанса с веб-сервером результат команды **show ip nat statistics** покажет увеличение количества совпадений до 5.



Настройка динамического NAT

Настройка динамического NAT

Задайте пул публичных IPv4-адресов от 209.165.200.241 до 209.165.200.250 с именем PUBLIC-POOL.

```
R2(config)# ip nat pool PUBLIC-POOL 209.165.200.241  
209.165.200.250 netmask 255.255.255.224
```

Настройте ACL 2, разрешающий преобразование NAT для устройств сети 192.168.10.0/24.

```
R2(config)# access-list 2 permit 192.168.10.0 0.0.0.255
```

Свяжите PUBLIC-POOL с ACL 2.

```
R2(config)# ip nat inside source list 2 pool PUBLIC-POOL
```

Настройка надлежащего внутреннего NAT-интерфейса.

```
R2(config)# interface Serial0/0/0
```

```
R2(config-if)# ip nat inside
```

Настройка надлежащего внешнего NAT-интерфейса.

```
R2(config)# interface Serial0/1/0
```

```
R2(config-if)# ip nat outside
```

Вы успешно настроили динамическое NAT.

Настройка динамического NAT

Проверка динамического NAT

```
R2# show ip nat translations
Pro Inside global      Inside local  Outside local  Outside global
--- 209.165.200.226    192.168.10.10 ---          ---
--- 209.165.200.227    192.168.11.10 ---          ---
R2#
R2# show ip nat translations verbose
Pro Inside global      Inside local  Outside local  Outside global
--- 209.165.200.226    192.168.10.10 ---          ---
      create 00:17:25, use 00:01:54 timeout:86400000, left
23:58:05, Map-Id(In): 1,
      flags:
none, use_count: 0, entry-id: 32, lc_entries: 0
--- 209.165.200.227    192.168.11.10 ---          ---
      create 00:17:22, use 00:01:51 timeout:86400000, left
23:58:08, Map-Id(In): 1,
      flags:
none, use_count: 0, entry-id: 34, lc_entries: 0
R2#
```

Команды **show ip nat translations** отображает все настроенные *статические преобразования адресов* и все *динамические преобразования*, созданные в результате обработки трафика.

Добавление ключевого слова **verbose** выводит дополнительную информацию о каждом преобразовании, включая *время, прошедшее после создания и использования записи*.

По умолчанию срок действия записей преобразования истекает через 24 часа, если настройка таймеров не была изменена с помощью команды **ip nat translation timeout timeout-seconds**.

Настройка динамического NAT

Проверка динамического NAT

```
R2# clear ip nat statistics
```

PC1 and PC2 establish sessions with the server

```
R2# show ip nat statistics
```

```
Total active translations: 2 (0 static, 2 dynamic; 0 extended)
Peak translations: 6, occurred 00:27:07 ago
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  Serial0/1/0
Hits: 24 Misses: 0
CEF Translated packets: 24, CEF Punted packets: 0
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool NAT-POOL1 refcount 2
  pool NAT-POOL1: netmask 255.255.255.224
  start 209.165.200.226 end 209.165.200.240
  type generic, total addresses 15, allocated 2 (13%), misses 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
R2#
```

Команда **show ip nat statistics** ВЫВОДИТ СВЕДЕНИЯ О

суммарном количестве активных преобразований,

параметрах конфигурации NAT,

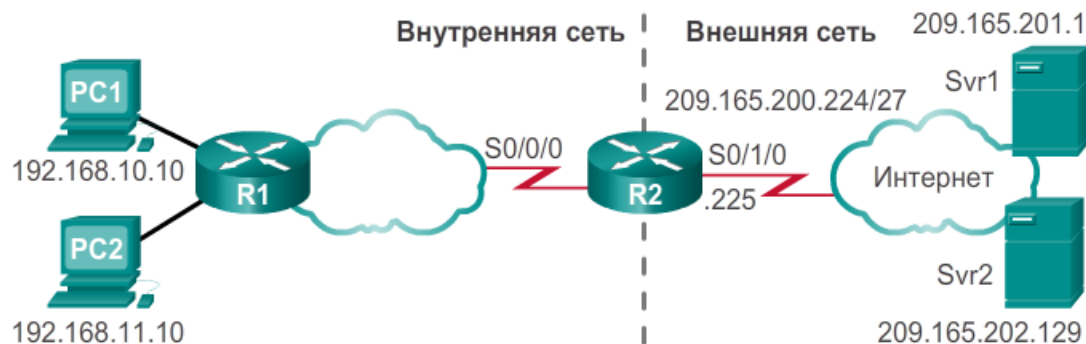
числе адресов в пуле и

числе выделенных адресов.

Настройка преобразования адреса и номера порта (PAT)

Настройка PAT: пул адресов

Пример PAT с пулом адресов



Задайте пул публичных IPv4-адресов с именем пула NAT-POOL2.

```
R2 (config) # ip nat pool NAT-POOL2 209.165.200.226  
209.165.200.240 netmask 255.255.255.224
```

Определите, какие адреса подходят для преобразования.

```
R2 (config) # access-list 1 permit 192.168.0.0 0.0.255.255
```

Привяжите NAT-POOL2 к ACL 1.

```
R2 (config) # ip nat inside source list 1 pool NAT-POOL2  
overload
```

Установите интерфейс serial 0/0/0 в качестве внутреннего интерфейса NAT.

```
R2 (config) # interface Serial0/0/0  
R2 (config-if) # ip nat inside
```

Установите интерфейс serial 0/1/0 в качестве внешнего интерфейса NAT.

```
R2 (config) # interface Serial0/1/0  
R2 (config-if) # ip nat outside
```

Ключевое слово **overload** задействует работу PAT.

Пример конфигурации создаёт преобразование с перегрузкой для пула NAT с именем NAT-POOL2.

NAT-POOL2 содержит адреса с 209.165.200.226 по 209.165.200.240.

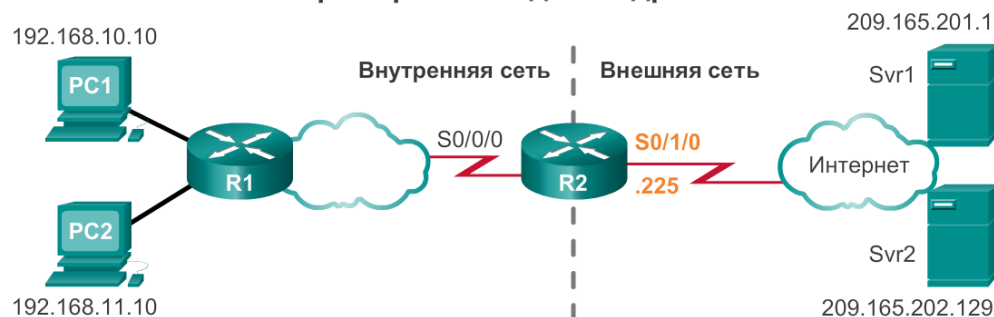
Объектами преобразования являются узлы сети 192.168.0.0/16.

В качестве внутреннего интерфейса определен интерфейс S0/0/0, а в качестве внешнего интерфейса — интерфейс S0/1/0.

Настройка преобразования адреса порта (PAT)

Настройка PAT для одного адреса

Пример PAT с одним адресом



- 1 Настройте ACL 1, разрешающий преобразование NAT для устройств сети 192.168.0.0/16.

```
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
```
- 2 Задайте внешний интерфейс serial 0/1/0 как внутренний глобальный адрес, который должен быть перегружен, с помощью ACL 1.

```
R2(config)# ip nat source list 1 interface serial 0/1/0 overload
```
- 3 Настройка надлежащего внутреннего NAT-интерфейса.

```
R2(config)# interface serial0/0/0  
R2(config-if)# ip nat inside
```
- 4 Настройка надлежащего внешнего NAT-интерфейса.

```
R2(config)# interface serial0/1/0  
R2(config-if)# ip nat outside
```

Вы успешно настроили PAT с помощью единого адреса.

Шаг 1	Задайте стандартный список доступа, разрешающий адреса, которые должны быть преобразованы. access-list <i>access-list-number</i> permit <i>source</i> [<i>source-wildcard</i>]
Шаг 2	Задайте динамическое преобразование адреса источника, указав ACL-список, выходной интерфейс и варианты перегрузки. ip nat inside source list <i>access-list-number</i> interface <i>type number</i> overload
Шаг 3	Задайте внутренний интерфейс. interface <i>type number</i> ip nat inside
Шаг 4	Задайте внешний интерфейс. interface <i>type number</i> ip nat outside

Настройка преобразования адреса и номера порта (PAT)

Проверка PAT

```
R2# show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	209.165.200.226:51839	192.168.10.10:51839	209.165.201.1:80	209.165.201.1:80
tcp	209.165.200.226:42558	192.168.11.10:42558	209.165.202.129:80	209.165.202.129:80

```
R2#
```

Двум различным внутренним узлам выделяется один и тот же внутренний глобальный IPv4-адрес 209.165.200.226.

Для различения этих двух транзакций в таблице NAT используются разные номера портов.

```
R2# clear ip nat statistics
```

```
R2# show ip nat statistics
```

```
Total active translations: 2 (0 static, 2 dynamic, 2 extended)
```

```
Peak translations: 2, occurred 00:00:05 ago
```

```
Outside interfaces:
```

```
Serial0/0/1
```

```
Inside interfaces:
```

```
Serial0/1/0
```

```
Hits: 4 Misses: 0
```

```
CEF Translated packets: 4, CEF Punted packets: 0
```

```
Expired translations: 0
```

```
Dynamic mappings:
```

```
-- Inside Source
```

```
[Id: 3] access-list 1 pool NAT-POOL2 refcount 2
```

```
pool NAT-POOL2: netmask 255.255.255.224
```

```
start 209.165.200.226 end 209.165.200.240
```

```
type generic, total addresses 15, allocated 1 (6%),  
misses 0
```

```
Total doors: 0
```

```
Appl doors: 0
```

```
Normal doors: 0
```

```
Queued Packets: 0
```

```
R2#
```

Команда `show ip nat statistics` позволяет проверить, что в пуле NAT-POOL2 выделен один адрес для обоих преобразований.

Результат выполнения команды содержит информацию о количестве и типе активных преобразований,

параметрах настройки NAT,

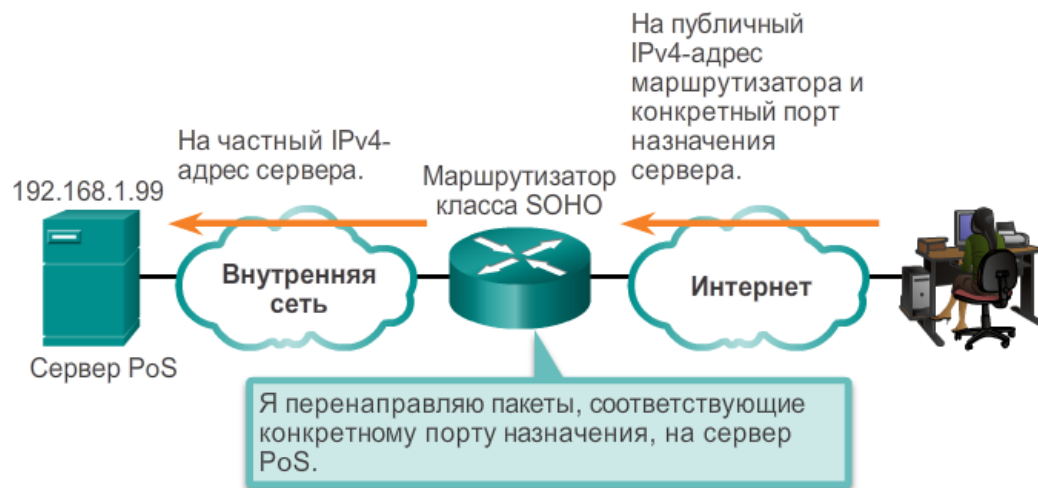
количестве адресов в пуле и

количестве выделенных адресов.

Переадресация портов

Переадресация портов

- Данный метод позволяет **внешним пользователям снаружи достигать порта для частного IPv4-адреса** (в локальной сети), используя маршрутизатор с поддержкой NAT.
- **Переадресация портов** (известная также как «проброс портов») — это процесс переадресации сетевого порта от одного узла сети на другой узел.
- Пакет, отправленный на публичный IP-адрес и порт маршрутизатора, может быть перенаправлен на частный IP-адрес и порт внутренней сети соответственно.
- Данный процесс полезен в случае, когда серверы имеют частные адреса, не доступные из внешних сетей.

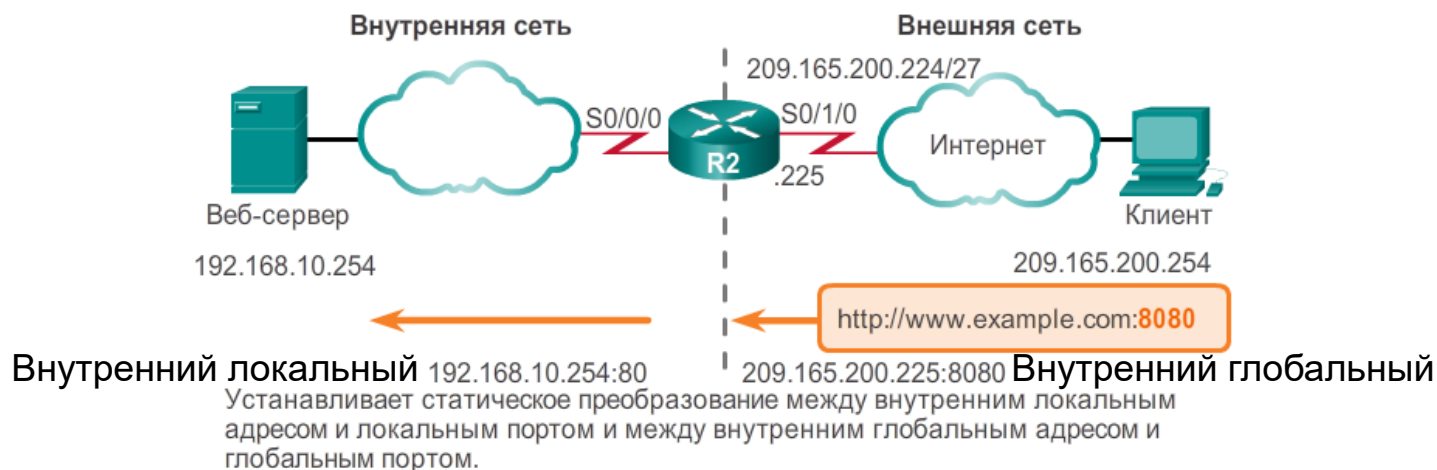


Владелец небольшого предприятия использует сервер PoS (пункт продаж) для отслеживания продаж и запасов на складе. Сервер доступен внутри склада, но поскольку ему назначен частный адрес IPv4, публичный доступ к этому серверу из Интернета невозможен. Включение на локальном маршрутизаторе переадресации портов предоставляет владельцу доступ к серверу пункта продаж из Интернета. **Переадресация портов на маршрутизаторе настраивается с помощью номера порта назначения и частного IPv4-адреса сервера пункта продаж.** Для доступа к серверу **клиентское ПО должно использовать публичный IPv4-адрес маршрутизатора и порта назначения сервера.**

Переадресация портов

Настройка переадресации портов в IOS

- В операционной системе IOS переадресация портов фактически является статическим NAT, выполняемым с заданным номером порта TCP или UDP.



```
R2(config)# ip nat inside source static tcp 192.168.10.254 80  
209.165.200.225 8080
```

Устанавливает интерфейс serial 0/0/0 в качестве внутреннего интерфейса NAT.

```
R2(config)# interface Serial0/0/0  
R2(config-if)# ip nat inside
```

Устанавливает интерфейс serial 0/1/0 в качестве внешнего интерфейса NAT.

```
R2(config)# interface Serial0/1/0  
R2(config-if)# ip nat outside
```



Настройка NAT IPv6

NAT для протокола IPv6.

- NAT позволяет обойти проблему недостаточного количества адресов в адресном пространстве IPv4.
- Протокол IPv6 с 128-битовым адресом предоставляет 340 ундециллионов адресов, поэтому вопрос нехватки адресов в адресном пространстве IPv6 не возникает.
- Протокол IPv6 разработан таким образом, что преобразование из публичного в частный адрес, актуальное для IPv4, оказывается излишним.
- Тем не менее IPv6 использует форму частной адресации, реализованную иначе, нежели в протоколе IPv4.
- В IPv6 NAT используется для обеспечения прозрачной коммуникации между протоколами IPv6 и IPv4.
- NAT64 предназначен для использования только как временное решение, в качестве механизма перехода.

Настройка NAT

Поиск и устранение неполадок в работе NAT. Команды show

```
R2# clear ip nat statistics
R2# clear ip nat translation *
R2#
```

Узел 192.168.10.10 подключается по протоколу telnet к серверу 209.165.201.1

```
R2# show ip nat statistics
Total active translations: 1 (0 static, 1 dynamic; 1 extended)
Peak translations: 1, occurred 00:00:09 ago
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  Serial0/0/0
Hits: 31 Misses: 0
CEF Translated packets: 31, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 5] access-list 1 pool NAT-POOL2 refcount 1
  pool NAT-POOL2: netmask 255.255.255.224
  start 209.165.200.226 end 209.165.200.240
  type generic, total addresses 15, allocated 1 (6%), misses 0
<output omitted>
R2# show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
tcp 209.165.200.226:19005 192.168.10.10:19005 209.165.201.1:23 209.165.201.1:23
R2#
```

```
R2# debug ip nat
IP NAT debugging is on
R2#
*Feb 15 20:01:311.670: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2817]
*Feb 15 20:01:311.682: NAT*: s=209.165.201.1, d=209.165.200.226->192.168.10.10 [4180]
*Feb 15 20:01:311.698: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2818]
*Feb 15 20:01:311.702: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2819]
```

Выполните следующие действия, чтобы убедиться, что NAT работает должным образом:

Шаг 1. В зависимости от конфигурации четко определите цели и задачи NAT. На данном этапе вы можете выявить проблему, связанную с конфигурацией.

Шаг 2. С помощью команды **show ip nat translations** убедитесь, что таблица преобразований содержит правильные преобразования.

Шаг 3. Используйте команды **clear** и **debug**, чтобы убедиться, что NAT работает должным образом. Проверьте, создаются ли динамические записи снова после их удаления.

Шаг 4. Подробно изучите, что происходит с преобразованным пакетом, и убедитесь, что маршрутизаторы используют правильные данные маршрутизации для передачи пакета.

Настройка NAT

Поиск и устранение неполадок в NAT с помощью команды debug

```
R2# debug ip nat
IP NAT debugging is on
R2#
*Feb 15 20:01:311.670: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2817]
*Feb 15 20:01:311.682: NAT*: s=209.165.201.1, d=209.165.200.226->192.168.10.10 [4180]
*Feb 15 20:01:311.698: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2818]
*Feb 15 20:01:311.702: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2819]
*Feb 15 20:01:311.710: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2820]
*Feb 15 20:01:311.710: NAT*: s=209.165.201.1, d=209.165.200.226->192.168.10.10 [4181]
*Feb 15 20:01:311.722: NAT*: s=209.165.201.1, d=209.165.200.226->192.168.10.10 [4182]
*Feb 15 20:01:311.726: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2821]
*Feb 15 20:01:311.730: NAT*: s=209.165.201.1, d=209.165.200.226->192.168.10.10 [4183]
*Feb 15 20:01:311.734: NAT*: s=192.168.10.10->209.165.200.226, d=209.165.201.1 [2822]
*Feb 15 20:01:311.734: NAT*: s=209.165.201.1, d=209.165.200.226->192.168.10.10 [4184]
output omitted
```

Результат показывает, что внутренний узел (192.168.10.10) создал трафик к внешнему узлу (209.165.201.1), и адрес источника был преобразован в адрес 209.165.200.226.

При расшифровке результатов отладки учитывайте значение перечисленных ниже символов:

- ***** (звездочка) — символ звездочки рядом с NAT показывает, что преобразование выполняется по пути с быстрой коммутацией. Коммутация первого пакета диалога всегда является программным процессом и поэтому выполняется медленнее. После появления в кэше соответствующей записи, остальные пакеты проходят по пути с быстрой коммутацией.
- **s=** — этот символ обозначает IP-адрес источника.
- **a.b.c.d--->w.x.y.z** — это значение показывает, что адрес источника a.b.c.d преобразуется в w.x.y.z.
- **d=** — этот символ обозначает IP-адрес назначения.
- **[xxxx]** — значение в скобках показывает идентификационный номер IP.