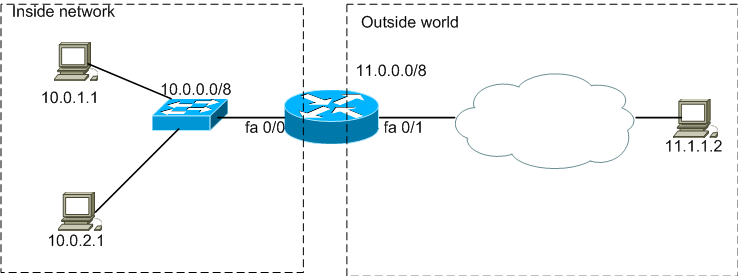
*NAT* (Network Address Translation) — технология трансляции сетевых адресов, т.е. подмены адресов в заголовке IP-пакета (иногда может еще и порт менять в заголовках TCP/UDP, но об этом позже).  
  
Другими словами, пакет, проходя через маршрутизатор, может поменять свой адрес источника и/или назначения.   
  
Зачем это нужно?  
1. Для обеспечения доступа из LAN, где чаще всего используются частные IP-адреса, в Internet, где маршрутизируются только глобальные IP-адреса.  
2. (*в меньшей степени*) для сокрытия топологии сети и создания некоторого защитного барьера для проникновения внутрь сети (обсудим это позже на примере).  
  
Классификация NAT:

1. *Static NAT* — статический NAT задает однозначное соответствие одного адреса другому. Иными словами, при прохождении через маршрутизатор, адрес(а) меняются на строго заданный адрес, один-к-одному. (к примеру 10.1.1.1 всегда заменяется на 11.1.1.1 и обратно, но никогда на 12.1.1.1). Запись о такой трансляции хранится неограниченно долго, пока есть строчка в конфиге.

2. *Dynamic NAT* — при прохождении через маршрутизатор, новый адрес выбирается динамически из некоторого куска адресов, называемого пулом (англ. pool). Запись о трансляции хранится некоторое время, чтобы ответные пакеты могли быть доставлены адресату. Если в течение некоторого времени трафик по этой трансляции отсутствует, трансляция удаляется и адрес возвращается в пул. Если требуется создать трансляцию, а свободных адресов в пуле нет, то пакет отбрасывается. Иными словами, хорошо бы, чтобы число внутренних адресов было ненамного больше числа адресов в пуле, иначе высока вероятность проблем с доступом наружу.

3. *Dynamic NAT with overload или PAT*. Работает почти также, как dynamic NAT, но при этом происходит трансляция много-в-один, задействуя при этом возможности транспортного уровня. Об этом подробнее на примере дальше.

**1. Inside source NAT**

Допустим у нас есть такая топология:  
  
Другими словами,   
а) подсеть внутренних адресов — 10.0.0.0/8  
б) подсеть внешних адресов — 11.0.0.0/8  
  
и мы хотим транслировать каким-то образом внутренние адреса во внешние при прохождении трафика через маршрутизатор.

**Что для этого нужно?**

1. Мы явно указываем, что мы хотим транслировать. Т.е. какой трафик и от каких хостов.  
2. Мы явно указываем, во что мы хотим транслировать, т.е. пул внешних адресов (или единственный адрес для статической трансляции).  
3. Помечаем внутренний и внешний интерфейс.  
4. Включаем трансляцию.  
  
На п.3 я себе позволю остановиться подробнее, потому что именно здесь часто происходит непонимание.

**Как это работает?**

Итак, допустим мы решили, что будем транслировать всю 10ю сеть целиком в 11ю. Задали их соответствующим образом (настройки потом, сначала теория). И пометили наши интерфейсы как внутренний (inside) и внешний (outside).  
Теперь, давайте разберемся, что делает именно *inside source NAT*. На самом деле, в названии зашита половина действия, а именно: у пакета, пришедшего на inside интерфейс меняется source. Но помните, мы говорили о том, что ответные пакеты должны доходить до нашего внутреннего хоста? Отсюда вторая половина действия: у пакета, пришедшего на outside, интерфейс меняется destination.  
  
Рассмотрим прямую трансляцию.

1. Трафик, приходя на интерфейс, помеченный как inside, если он соответствует тому, что мы хотим транслировать, маркируется как *возможно\_транслируемый*. Часто полагают, что в этот момент происходит трансляция, но это не так.

2. Следующим этапом, трафик подвеграется маршрутизации (PBR и обычной). И если при этом трафик направляется на интерфейс, помеченный как outside — только тогда происходит трансляция. Если трансляция динамическая, маршрутизатор проверяет ее наличие в таблице трансляций. Если ее там нет — создает, если уже есть — обнуляет счетчик неактивности. Если же пакет попадает на выход на интерфейс, не помеченный как outside — трансляция НЕ происходит.  
  
Теперь обратная трансляция.

1. Трафик, попадая на outside интерфейс, в противовес прямой трансляции, сначала подвергается NAT. Если трансляция существует (неважно, динамическая или статическая), в случае с inside source NAT, у него меняется destination. И только после этого трафик подвергается маршрутизации и перенаправляется по назначению.  
  
Поэтому маркировать интерфейсы как inside или outside нужно именно принимая во внимание механизм работы.

Замечания и следствия.

1. Для обратной трансляции не обязательно наличие метки inside на каком-либо интерфейсе. Все равно, если прямая трансляция существует, обратная трансляция сработает до маршрутизации. Но когда будет существовать такая трансляция, ведь мы обсуждали, что трафик должен пройти через inside интерфейс для создания прямой трансляции?

Отсюда

2. Трафик самого роутера подвергается трансляции, если он попадает на интерфейс, помеченный как outside и удовлетворяет правилу NAT. И это сколь полезно, столь и опасно. С одной стороны, мы можем транслировать трафик роутера как и любой другой. С другой стороны, многие хотят описать трафик, подлежащий трансляции как *permit any*, но тогда и, например, пакеты протоколов маршрутизации будут транслироваться, что приведет к сбою.  
3. Интерфейсы типа loopback маршрутизатора трактуются как и любые другие, мы можем метить их как inside или outside, заворачивать на них трафик и получать от этого профит :)  
  
Теперь посмотрим общую конфигурацию, а потом еще несколько частных случаев.

**Конфигурация inside source NAT**

**inside source dynamic NAT**

1. Указываем, что транслировать. Для этого создаем access-list, перечисляющий трафик. Например, в нашем случае достаточно одной строчки:  
(config)# access-list 100 permit ip 10.0.0.0 0.255.255.255 any  
Замечание. В ACL могут встречаться строчки deny. Вопреки распространенному заблуждению, трафик удовлетворяющей данной строчке не дропается, а просто не подвеграется трансляции. Так же, ACL может быть стандартным и расширенным, нумерованным и именованным.  
2. Создаем пул из адресов, указывая стартовый и конечный адрес. Например так:  
(config)# ip nat pool NAME\_OF\_POOL 11.1.1.10 11.1.1.20 netmask 255.255.255.0  
Замечания.  
1. Стартовый и конечный адрес в пуле могут совпадать, тогда трансляция будет в 1 адрес.  
2. Опция netmask, хотя и является обязательной, по моему мнению — рудимент. Она позволяет вырезать из диапазона адресов в пуле те адреса, которые являются адресами подсети или бродкастными при данной маске.  
3. Маркируем интерфейсы. В нашем случае достаточно  
(config)# interface fa 0/0  
(config-if)# ip nat inside  
и   
(config)# interface fa 0/1  
(config-if)# ip nat outside  
  
4. создаем собственно трансляцию:   
ip nat inside source list 100 pool NAME\_OF\_POOL   
  
вуаля :) Если мы теперь обратимся например с хоста 10.1.1.1 к хосту 11.1.1.2, то получим такую трансляцию:  
Router#sh ip nat translations   
Pro Inside global Inside local Outside local Outside global  
tcp 11.1.1.10:55209 10.0.1.1:55209 11.1.1.2:23 11.1.1.2:23  
  
Интересно, что хотя в таблице явно записаны source port и destination port, трансляция создается целиком для адреса. И на время ее жизни в таблице трансляция, пакеты снаружи могут проходить на внешний адрес (inside global)  
Например, пинг с некоторого адреса во внешней сети на наш inside global будет успешным (на время жизни трансляции):  
R4#ping 11.1.1.10  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 11.1.1.10, timeout is 2 seconds:  
!!!!!  
Иными словами, открывается трансляция единожды и к некоторому хосту, после этого некоторое время действует для любого адреса извне.

**inside source dynamic NAT with overload**

П. 1,2 и 3 — как в предыдущем разделе.  
4. Создаем собственно трансляцию:   
ip nat inside source list 100 pool NAME\_OF\_POOL **overload**  
Видим, что добавилось всего одно слово: overload. Но оно существенно изменило схему работы трансляции.  
Как было сказано, PAT — это трансляция много-в-мало или даже много-в-один. Но чтобы можно было отличить трафик одного соединения от другого, маршрутизатор будет менять не только IP-адреса, но еще и TCP/UDP порты.  
Замечание. Схема работы с портами (когда меняется source, когда destination) — такая же, как и схема работы с IP-адресами.  
Другими словами, при обращении изнутри наружу меняется source IP и source port, запись об этом вносится в таблицу трансляций. При обратной трансляции — все меняется наоборот.   
  
Посмотрим, что изменилось:  
R3#sh ip nat translations   
Pro Inside global Inside local Outside local Outside global  
tcp 11.1.1.11:21545 10.0.1.1:21545 11.1.1.2:23 11.1.1.2:23  
tcp 11.1.1.11:49000 10.0.2.1:49000 11.1.1.2:23 11.1.1.2:23  
Видим, что разные внутренние адреса (10.0.1.1 и 10.0.2.1) странслировались в один внешний (11.1.1.11).   
  
Замечания.  
1. Кажется, что source-port не был изменен, как обещали, непорядок :). На деле, маршрутизатор пытается сохранить source port всеми доступными средствами. В частности, если порт inside global адреса уже был занят, он возьмет следующий адрес в пуле и проверит его порт на занятость. И только не найдя адреса со свободным портом возьмет следующий свободный.  
2. Поведение такой трансляции отличается от поведения обычного dynamic NAT еще и тем, что доступ снаружи на inside global адрес невозможен. Именно это я имел ввиду, когда говорил о некоторой повышенной безопасности при использовании PAT, т.к. фактически все соединения инициируются изнутри нашей сети, а снаружи нам могут приходить только ответы на них.  
3. Если мы получили у провайдера не целый блок адресов, а один несчастный адрес, который тут же и назначили внешнему интерфейсу маршрутизатора, можно не городить огород с пулом в один адрес, а сразу писать например так:  
(config)# ip nat inside source list 100 interface fa0/1 overload

**inside source static NAT and PAT**

Много упоминалось о статических трансляциях, давайте наконец их обсудим.  
  
Зачем это нужно?  
Мы обсудили, что если в случае dynamic NAT трансляция не создана и в случае PAT, доступ извне невозможен. Если даже в случае dynamic NAT трансляция создана, то inside global адрес может меняться. И обращаться к нашему внутреннему хосту по какому-то внешнему адресу невозможно.  
Тем не менее, нередки ситуации, когда внутри корпоративной сети есть сервер, доступ которому извне по статическому внешнему адресу жизненно необходим. В таком случае, можно выставить его прямиком в Интернет, назначив глобальный адрес. Но часто это не очень удобно, например по соображениям безопасности. И в таких случаях нам на помощь приходит static NAT.  
  
Он создает двустороннюю и постоянную трансляцию. Так что наш хост всегда будет доступен по одному внешнему адресу и эта трансляция никогда не вылетит из таблицы трансляций по таймауту.  
собственно настройка.  
Сразу создаем трансляцию:  
(config)# ip nat inside source static 10.0.1.1 11.1.1.21  
Маркируем интерфейсы и вуаля!  
R3#sh ip nat translations   
Pro Inside global Inside local Outside local Outside global  
icmp 11.1.1.21:14 10.0.1.1:14 11.1.1.2:14 11.1.1.2:14  
--- 11.1.1.21 10.0.1.1 --- ---  
Как видим, появилось две записи — одна постоянная, другая (чисто информативная) — временная, вызванная трафиком изнутри наружу.  
Замечание. Появление таких информативных записей можно отключить командой  
(config)# no ip nat create flow-entries   
  
Идем дальше. Часто бывает, что нужно выставить наружу не целый адрес, а только один порт (например 80й для www-сервера). Никаких проблем, можно создать и статическую PAT-трансляцию для некоторых портов:  
(config)# ip nat inside source static tcp 10.0.1.1 80 11.1.1.21 80  
(config)# ip nat inside source static udp 10.0.1.1 5060 11.1.1.21 7877   
Видим, что порты одного и того же внешнего адреса можно пробрасывать на разные порты внутренних, и управлять трансляцией портов при этом тоже возможен.  
  
В заключение добавлю, что изменять различные таймауты для NAT можно командой  
Router(config)#ip nat translation ?   
arp-ping-timeout Specify timeout for WLAN-NAT ARP-Ping  
dns-timeout Specify timeout for NAT DNS flows  
finrst-timeout Specify timeout for NAT TCP flows after a FIN or RST  
icmp-timeout Specify timeout for NAT ICMP flows  
max-entries Specify maximum number of NAT entries  
port-timeout Specify timeout for NAT TCP/UDP port specific flows  
pptp-timeout Specify timeout for NAT PPTP flows  
routemap-entry-timeout Specify timeout for routemap created half entry  
syn-timeout Specify timeout for NAT TCP flows after a SYN and no  
further data  
tcp-timeout Specify timeout for NAT TCP flows  
timeout Specify timeout for dynamic NAT translations  
udp-timeout Specify timeout for NAT UDP flows

ВАРИАНТ 2

NAT (Network Address Translation) — трансляция сетевых адресов, технология, которая позволяет преобразовывать (изменять) IP адреса и порты в сетевых пакетах.

NAT используется чаще всего для осуществления доступа устройств из сети предприятия(дома) в Интернет, либо наоборот для доступа из Интернет на какой-либо ресурс внутри сети.

Сеть предприятия обычно строится на частных IP адресах. Согласно RFC 1918 под частные адреса выделено три блока:

10.0.0.0 — 10.255.255.255  (10.0.0.0/255.0.0.0  (/8))  
172.16.0.0 — 172.31.255.255  (172.16.0.0/255.240.0.0  (/12))  
192.168.0.0 — 192.168.255.255  (192.168.0.0/255.255.0.0   (/16))

Эти адреса не маршрутизируются в Интернете, и провайдеры должны отбрасывать пакеты с такими IP адресами отправителей или получателей.

Для преобразования частных адресов в Глобальные (маршрутизируемые в Интернете) применяют NAT.

Помимо возможности доступа во внешнюю сеть (Интернет), NAT имеет ещё несколько положительных сторон. Так, например, трансляция сетевых адресов позволяет скрыть внутреннюю структуру сети и ограничить к ней доступ, что повышает безопасность. А ещё эта технология позволяет экономить Глобальные IP адреса, так как под одним глобальным адресом в Интернет может выходить множество хостов.

Настройка NAT на маршрутизаторах Cisco под управлением IOS включает в себя следующие шаги

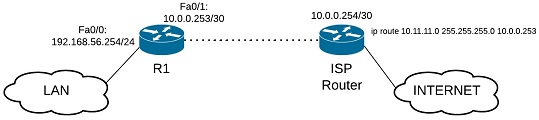
1.  Назначить внутренний (Inside) и внешний (Outside) интерфейсы

2. Определить для кого (каких ip адресов) стоит делать трансляцию.

3. Выбрать какой вид трансляции использовать

4. Осуществить проверку трансляций

Пусть, для примера, у нас будет роутер R1 с двумя интерфейсами.



FastEthernet0/0 — подключен к локальной сети 192.168.56.0/24 и имеет IP адрес: 192.168.56.254/24

FastEthernet0/1 — подключен к внешней сети (Провайдеру) и имеет IP адрес: 10.0.0.253/30 (В данном примере во внешней сети используется тоже частный адрес, так как это только пример, в реальной же ситуации у Вас скорее всего будет использоваться Глобальный адрес)

**1.** Внутренним интерфейсом обычно выступает тот, к которому подключена локальная сеть. Внешним — к которому подключена внешняя сеть, например сеть Интернет провайдера.

Таким образом FastEthernet0/0 — это Inside интерфейс, а FastEthernet0/1 — Outside

Конфигурируем интерфейсы:

R1(config)#interface fastEthernet 0/0  
R1(config-if)#ip nat inside  
R1(config-if)#exit  
R1(config)#

R1(config)#interface fastEthernet 0/1  
R1(config-if)#ip nat outside  
R1(config-if)#exit  
R1(config)#

**2.**Трансляцию будем делать для всей локальной сети 192.168.56.0/24

Для этого создаём ACL:

R1(config)#access-list 1 permit 192.168.56.0 0.0.0.255  
R1(config)#

**3.**  Существует три вида трансляции Static NAT, Dynamic NAT, Overloading.

**Static NAT** — Статический NAT, преобразование IP адреса один к одному, то есть сопоставляется один адрес из внутренней сети с одним адресом из внешней сети.

 Предположим, что провайдер на нас маршрутизирует сеть 10.11.11.0/24, таким образом предоставив не один, а 255 IP адресов. И мы хотим, что бы из внешней сети по адресу 10.11.11.10 был доступен наш внутренний сервер 192.168.56.10, тогда следует ввести следующую команду:

R1(config)#ip nat inside source static 192.168.56.10 10.11.11.10  
R1(config)#

Таким образом происходит подмена внутреннего адреса источника 192.168.56.10 на внешний адрес 10.11.11.10 при прохождении пакета из внутренней сети во внешнюю. При обратном следовании пакета произойдёт подмена внешнего IP адреса назначения 10.11.11.10 на внутренний 192.168.56.10. Все сервисы, которые запущены на внутреннем сервере 192.168.56.10 доступны из внешней сети при обращении на адрес 10.11.11.10

Если нет необходимости открывать наружу все порты (сервисы), то можно ограничиться и определёнными, например:

R1(config)#ip nat inside source static tcp 192.168.56.10 22 10.11.11.10 2222  
R1(config)#

 В этом случае при обращении из внешней сети на адрес 10.11.11.10 и tcp порт 2222 произойдет соединение с 22-м tcp портом внутреннего сервера 192.168.56.10

**Dynamic NAT** — Динамический NAT, преобразование внутреннего адреса/ов в один из группы внешних адресов. Перед использованием динамической трансляции, нужно задать nat-пул внешних адресов

R1(config)#  
R1(config)#ip nat pool GLOBALPOOL 10.11.11.20 10.11.11.100 netmask 255.255.255.0  
R1(config)#ip nat inside source list 1 pool GLOBALPOOL  
R1(config)#

В этом случае адреса из сети 192.168.56.0/24, при обращении к внешней сети, будут преобразовываться в адреса из пула GLOBALPOOL, пока он не закончится. Если не останется в пуле свободных адресов — то трансляции выполняться не будут. То есть в нашем примере одновременно во внешнюю сеть выйдут только 80 адресов из сети 192.168.56.0/24, остальным придётся ждать пока не закончится какая-нибудь трансляция.

**Overloading** — позволяет преобразовывать несколько внутренних адресов в один внешний. Для осуществления такой трансляции используются порты, поэтому иногда такой NAT называют PAT (Port Address Translation).  С помощью PAT можно преобразовывать внутренние адреса во внешний адрес, заданный через пул или через адрес на внешнем интерфейсе.

через пул:

R1(config)#  
R1(config)#ip nat pool GLOBALPOOL 10.11.11.3 10.11.11.3 netmask 255.255.255.0  
R1(config)#ip nat inside source list 1 pool GLOBALPOOL overload  
R1(config)#

 Здесь все внутренние адреса из сети 192.168.56.0/24 (описанные в access-list 1 выше) будут преобразованы во внешний адрес 10.11.11.3

через внешний интерфейс:

R1(config)#  
R1(config)#ip nat inside source list 1 interface fastEthernet 0/1 overload  
R1(config)#

 Все внутренние адреса 192.168.56.0/24 будут преобразованы в адрес на интерфейсе fastEthernet0/1 (напомню, что он в нашем примере 10.0.0.253/30)

**4.** Посмотреть существующие трансляции можно командой «show ip nat translations». Отладка запускается командой «debug ip nat»

R1#  
R1#show ip nat translations  
Pro Inside global Inside local Outside local Outside global  
udp 10.0.0.253:60672 192.168.56.10:60672 8.8.8.8:53 8.8.8.8:53  
tcp 10.0.0.253:59235 192.168.56.10:59235 74.125.143.94:80 74.125.143.94:80  
R1#

Пример конфигурации роутера с настроенным NAT overloading:

R1#sh run  
Building configuration... Current configuration : 1163 bytes  
!  
! Last configuration change at 01:03:44 YST Sun Mar 31 2013 by cisco  
! NVRAM config last updated at 01:03:46 YST Sun Mar 31 2013 by cisco  
!  
version 12.3  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname R1  
!  
boot-start-marker  
boot-end-marker  
!  
enable secret 5 $1$FOp2$9G9FvXT/lRo1HqKkgwVWv/  
!  
clock timezone YST 10  
aaa new-model  
!  
!  
aaa authentication login default local  
aaa session-id common  
ip subnet-zero  
ip cef  
!  
!  
!  
ip domain name test.ru  
ip audit po max-events 100  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
username cisco secret 5 $1$jOx6$M96EAbwK4qcWtGlxHLt3J/  
!  
!  
!  
!  
!  
!  
interface FastEthernet0/0  
ip address 192.168.56.254 255.255.255.0  
ip nat inside  
duplex auto  
speed auto  
!  
interface FastEthernet0/1  
ip address 10.0.0.253 255.255.255.252  
ip nat outside  
duplex auto  
speed auto  
!  
ip nat inside source list 1 interface FastEthernet0/1 overload  
ip classless  
ip route 0.0.0.0 0.0.0.0 10.0.0.254  
!  
no ip http server  
no ip http secure-server  
!  
access-list 1 permit 192.168.56.0 0.0.0.255  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
exec-timeout 60 0  
transport input telnet ssh  
!  
end  
R1#