# ASA 8.3 NAT

О, сколько нам открытий чудных…
Пушкин А.С.

**Команды NAT для ОС Cisco ASA версии 8.3**

Произошли большие изменения в работе NAT на ASA, начиная с версии OS 8.3. В этой статье я попытаюсь сделать достаточно подробный обзор этих изменений с примерами из прошлых версий. Для тех, кто до сих пор работал на ASA (а может быть даже и на PIX) с версиями более ранними, будет проще понять новые команды NAT, видя перед глазами старые хорошо изученные примеры. Переход на новые версии все равно неизбежен. Всё чаще стали встречаться ASA с новой версией операционной системы, поэтому лучше быть готовым к этому сейчас. В написании этой статьи я опирался на официальные документы Cisco.

Во-первых, это Руководство по CLI 8.3, раздел «Конфигурация NAT» (http://www.cisco.com/en/US/docs/security/asa/asa83/configuration/guide/nat_overview.html )

Во-вторых, это Руководство по миграции ASA серии 5500 на версию 8.3 и выше (http://www.cisco.com/en/US/docs/security/asa/asa83/upgrading/migrating.html )

Какие же основные изменения произошли с появлением версии 8.3?

- Использование реальных (нетранслированных) IP-адресов в ACL, но не повсеместно, а только в определенном функционале, например в NAT. Но об этом я расскажу в другой раз.
- NAT – этот функционал был полностью переделан, чтобы увеличить гибкость и функциональность. Соответственно, все команды связанные с NAT были изменены.

Начнем рассмотрение изменений с последнего пункта, т.к. понимание всего остального придет с пониманием новых команд NAT. Замечу только, что обновление ASA с версий более ранних на версию 8.3 не проходит безболезненно. Старая конфигурация при обновлении довольно сильно изменяется, происходит автоматическая конвертация правил NAT и связанных с ним ACL в новый формат. Создаются объекты для использования с NAT, некоторые ACL перестают работать. В общем новый конфиг после перезагрузки вам может показаться какой-то тарабарщиной. Да к тому же некоторые сетевые сервисы в вашей сети могут перестать функционировать. Поэтому я бы не рекомендовал обновлять боевые устройства до версии 8.3, тщательно не подготовив новый конфиг. А конфиг лучше всего обкатывать на тестовом полигоне.

В версии 8.3 весь функционал NAT делится на два типа:

- **Network Object NAT** – команда NAT дается внутри сетевого объекта (object network), описывающего реальные адреса сети, диапазона сети или хоста, которые будут транслироваться. Правило NAT является параметром этого объекта и применяется или к источнику или к получателю пакета. **Regular NAT, Auto NAT** – синонимы **Network Object NAT**.
- **Twice NAT** – команда NAT дается в глобальном окружении. Сетевые объекты или группы объектов, описывающие реальные и транслируемые адреса, являются частью правила NAT. Одно правило может описывать как трансляцию адресов источника, так и трансляцию адресов получателя пакетов. **Policy NAT**, **Manual NAT** – это все синонимы **Twice NAT**.

Начиная с версии 8.3 больше нет команд:

- `global`
- `nat (в её старом виде)`

- `nat-control`
- `static`

Как обойтись без них? Давайте посмотрим.

**Regular Dynamic PAT**

Предлагаю начать с самого простого примера. У нас есть локальная сеть 10.1.2.0/24, которую надо выпустить в Интернет. У нас есть ASA с двумя интерфейсами (внутренним и внешним), на внешнем интерфейсе указан один адрес, который нам дал провайдер (Рисунок 1).

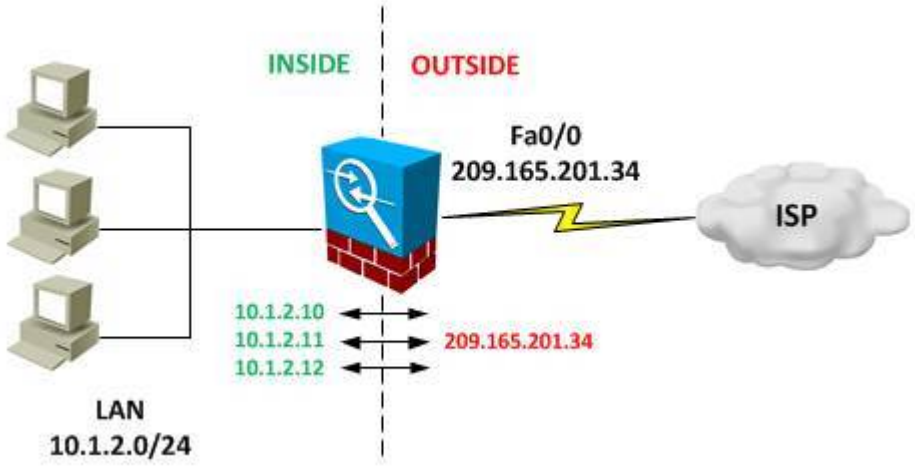


Рисунок 1

Получается, что нужно делать PAT, чтобы хосты из локальной сети могли бороздить просторы Интернет.

Так это выглядело раньше до версии 8.3:

```
nat (inside) 1 10.1.2.0 255.255.255.0
global (outside) 1 interface
```

В новой интерпретации это выглядит так:

```
object network myLAN
   subnet 10.1.2.0 255.255.255.0
   nat (inside,outside) dynamic interface
```

Эти команды как раз являются примером Network Object NAT. Мы описали сетевой объект, адреса которого надо транслировать и здесь же указали правило NAT. Вы скажете, что записи в конфигурации стали длиннее. Для многих, привыкших к командам NAT в старом формате, это еще и непонятно. Но я попытаюсь доказать обратное. Стало понятнее и логичнее. Дело только в привычке. Да, теперь чтобы транслировать адреса необходимо создавать объекты (object network или object service) и группы объектов (object-group), но эти объекты и группы можно также использовать в ACL или даже в Twice NAT.

Есть несколько хитростей в понимании команд NAT в версии 8.3 и выше.

В написании команды NAT, первая её часть  nat(inside,outside) подсказывает нам как проходит трафик, который мы транслируем. Первый интерфейс, указанный в скобках, это интерфейс через который оригинальный трафик попадает в ASA. Второй интерфейс – это интерфейс, через который транслированный трафик покидает ASA. Пока логика такая же как и раньше.

Команда interface, которая в нашем примере указана в правиле NAT после команды dynamic, всегда ссылается на второй интерфейс, указанный в скобках, т.е. в нашем случае на outside. Когда мы создаем правило для PAT, используя IP-адрес интерфейса, рекомендуется указывать не IP-адрес этого интерфейса, а ключевое слово interface.

Таким образом запись nat (inside,outside) dynamic interface можно прочитать так: «Подвергать трансляции весь трафик из сегмента INSIDE, который направляется в сегмент OUTSIDE, в IP-адрес, который присвоен интерфейсу outside». И, конечно же, ответный трафик подвергать обратной трансляции. Т.к. команда nat в данном примере является параметром команды object network, которая описывает сеть 10.1.2.0/24, то именно эти адреса и будут транслироваться в адрес внешнего интерфейса.

Если у вас в LAN большое количество подсетей и со всех адресов разрешено выходить в Интернет, то можно поступить проще и любые адреса из локальной сети транслировать в адрес внешнего интерфейса. Вот как это было:

```
nat (inside) 1 0 0
global (outside) 1 interface
```

И вот как стало выглядеть сейчас:

```
object network anyLAN
   subnet 0.0.0.0 0.0.0.0
   nat (inside,outside) dynamic interface
```

Если у нас в сети есть еще сегменты, например DMZ, то трафик, который идет из сегмента INSIDE в DMZ и обратно не подвергается никакой трансляции, пока мы не укажем правило трансляции для этого трафика. В этом случае между этими двумя сегментами работает только маршрутизация, т.е. теперь функциональность ***nat control*** выключена безвозвратно, а столько было вокруг неё разговоров.

**Regular Dynamic NAT**

Теперь немного усложним задачу. Предположим, что провайдер нам выдал не один IP-адрес, а некоторый диапазон, например, 209.165.201.32/27. И мы из этого диапазона решили выделить 20 IP-адресов для трансляции приватных адресов нашей локальной сети (Рисунок 2).

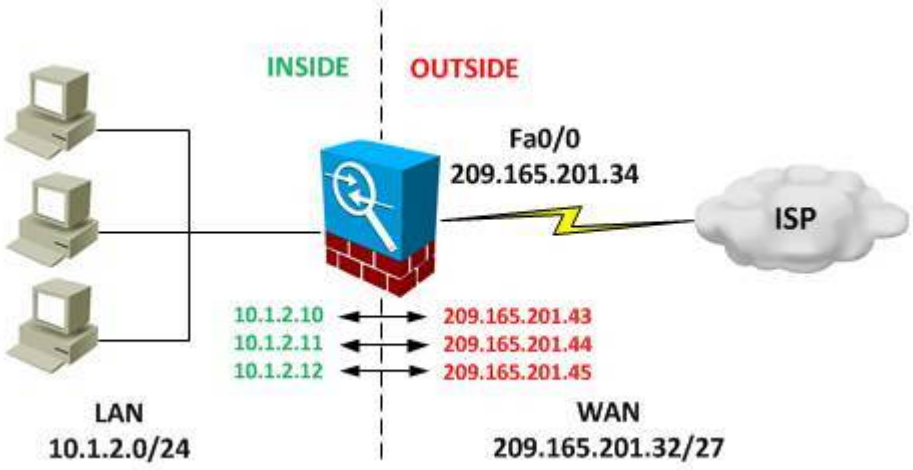


Рисунок 2

В старых версиях это выглядело следующим образом:

```
nat (inside) 1 10.1.2.0 255.255.255.0
global (outside) 1 209.165.201.43-209.165.201.62
```

Начиная с версии 8.3, то же самое выглядит так:

```
object network NAT-pool
   range 209.165.201.43 209.165.201.62
object network myLAN
   subnet 10.1.2.0 255.255.255.0
   nat (inside,outside) dynamic NAT-pool
```

Прекрасно, работает! И тут мы решили, что адрес, который указан на внешнем интерфейсе, мы будем использовать для PAT, на тот случай, если одновременно должны создаваться более 20 трансляций в Интернет. В старом варианте это бы выглядело так:

```
nat (inside) 1 10.1.2.0 255.255.255.0
global (outside) 1 209.165.201.43-209.165.201.62
global (outside) 1 interface
```

В новом так:

```
object network NAT-pool
   range 209.165.201.43 209.165.201.62
object network myLAN
   subnet 10.1.2.0 255.255.255.0
   nat (inside,outside) dynamic NAT-pool interface
```

Вот так легко и просто соединяются Dynamic NAT и interface PAT в одном правиле. Кстати, можно было бы добавить еще один IP-адрес для PAT. Тогда, после того как исчерпался пул адресов NAT, стал бы использоваться PAT через этот адрес и только после того как и он исчерпал свои возможности, только тогда заработал бы PAT через внешний интерфейс. Вот как это можно было сделать до версии 8.3:

```
nat (inside) 1 10.1.2.0 255.255.255.0
global (outside) 1 209.165.201.43-209.165.201.62
```

```
global (outside) 1 209.165.201.42
global (outside) 1 interface
```

Начиная с версии 8.3, чтобы сделать то же самое, нам необходимо сначала создать два сетевых объекта и объединить их в одну группу, а затем сослаться на эту группу в правиле NAT:

```
object network NAT-pool
   range 209.165.201.43 209.165.201.62
object network PAT-IP
   host 209.165.201.42
object-group network NAT-PAT-pool
   network-object object NAT-pool
   network-object object PAT-IP
object network myLAN
   subnet 10.1.2.0 255.255.255.0
   nat (inside,outside) dynamic NAT-PAT-pool interface
```

**Regular Static NAT**

Предположим, у нас есть сегмент DMZ, в котором находится сервер, и нам необходимо транслировать его IP-адрес в IP-адрес Интернет (Рисунок 3).

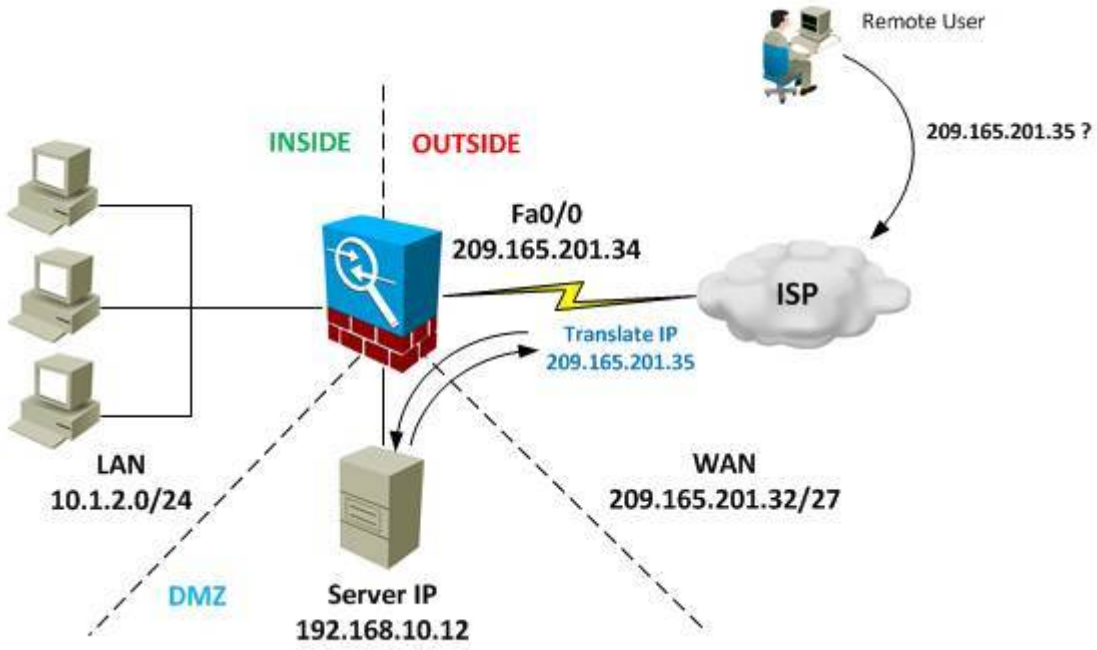


Рисунок 3

Раньше это делалось командой:

```
static (dmz,outside) 209.165.201.35 192.168.10.12 netmask
255.255.255.255
```

Теперь это будет выглядеть так:

```
object network DMZ-server
```

```
   host 192.168.10.12
   nat (dmz,outside) static 209.165.201.35
```

В версии 8.3 в команде nat можно указывать ключевое слово any, которое указывает на любой интерфейс, и позволяет делать трансляции не в один интерфейс, а сразу в несколько. Допустим, в DMZ есть сервер, адрес которого надо транслировать не только в OUTSIDE, но и в другие сегменты. Это можно сделать следующим образом:

```
object network DMZ-server
   host 192.168.10.12
   nat (dmz,any) static 209.165.201.35
```

Такая конфигурация позволяет не только из OUTSIDE попасть на сервер в DMZ, но и пользователям из сегмента INSIDE получить доступ к серверу в сегменте DMZ по адресу 209.165.201.35. Если трафик из INSIDE к этому адресу придет на ASA, она его транслирует в приватный адрес 192.168.10.12. Но надо помнить, что в режиме **transparent firewall** команду any использовать нельзя.

**Regular Static PAT with Port Translation**

В этом примере у нас будет два web-сервера в DMZ и один IP-адрес на внешнем интерфейсе ASA. Нам необходимо сделать проброс портов (Port Translation) для доступа к web-серверам из Интернет по разным портам (Рисунок 4).

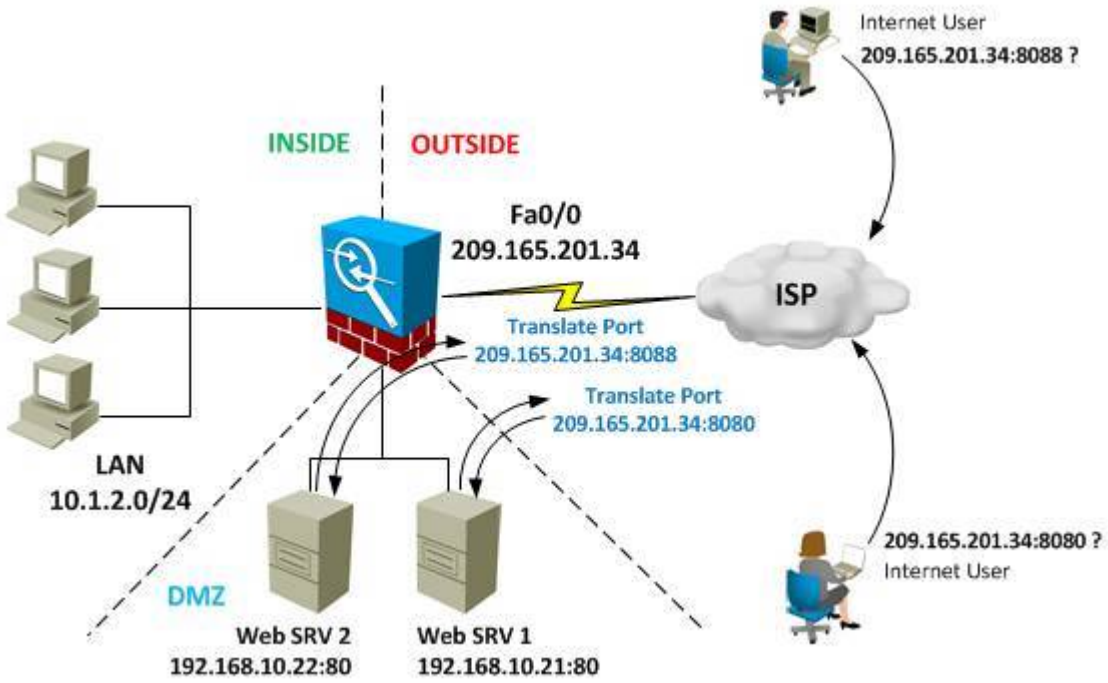


Рисунок 4

До появления версии 8.3 это настраивалось так:

```
static (dmz,outside) tcp interface 8080 192.168.10.21 www
```

```
static (dmz,outside) tcp interface 8088 192.168.10.22 www
```

Вот как это будет выглядеть в версии 8.3:

```
object network dmz-webserver1
   host 192.168.10.21
   nat (dmz,outside) static interface service tcp 8080 www
object network dmz-webserver2
   host 192.168.10.22
   nat (dmz,outside) static interface service tcp 8088 www
```

Таким образом, если из Интернет обратиться на IP-адрес внешнего интерфейса по порту 8080, то мы попадаем на 80-ый порт сервера 192.168.10.21 в DMZ. Если обратиться на этот же IP-адрес внешнего интерфейса, но уже на порт 8088, то произойдет соединение на порт 80 другого сервера – 192.168.10.22.

Все примеры, рассмотренные выше, относятся к **Network Object NAT**. Теперь пришло время рассмотреть **Twice NAT**.

**Twice NAT**

Как уже было сказано, Twice NAT позволяет в одном правиле указать как адреса источника, так и адреса назначения. Таким образом, с помощью Twice NAT мы можем составлять правила NAT, которые дают нам возможность транслировать адрес источника в *один* адрес, если мы посылаем пакеты серверу А, и в *другой* адрес, если мы посылаем пакеты серверу Б.

И для реальных и для транслированных адресов создаются сетевые объекты или группы объектов (object network /object-group network). Сетевые группы объектов обычно используются, чтобы описать транслируемый пул адресов, который содержит несколько диапазонов IP-адресов или несколько адресов хостов или подсетей.

Если мы создаем правило статического NAT с трансляцией портов, то необходимо создать сервисный объект (**object service**) с описанием этих TCP или UDP портов.

Давайте рассмотрим примеры. Начнем с простого.

**Dynamic Twice NAT**

Вспомним ситуацию, когда нам необходимо подвергать трансляции трафик, отвечающий некоторому условию. Например, весь ip-трафик, идущий из локальной сети к хосту 150.43.21.18 транслировать в адреса диапазона 209.165.201.10-209.165.201.20 (Рисунок 5).

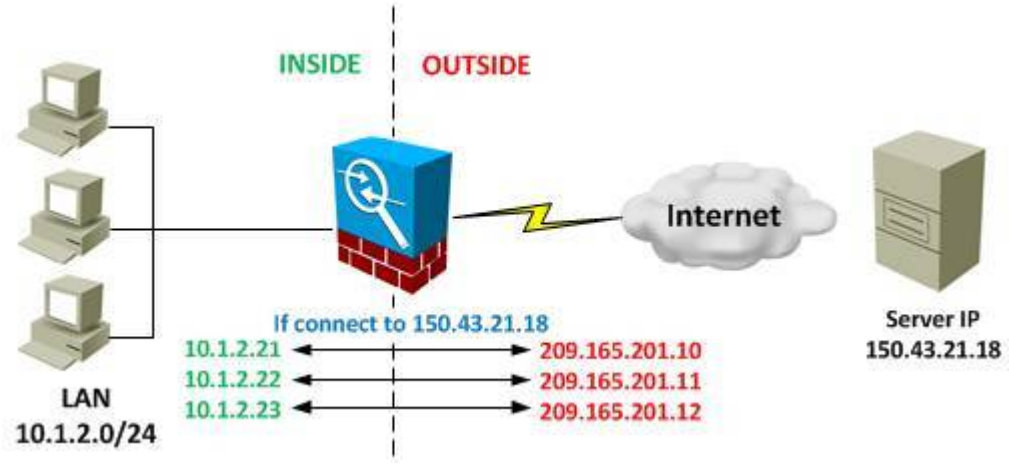


Рисунок 5

Вот как мы это делали раньше:

```
access-list 101 permit ip 10.1.2.0 255.255.255.0 host 150.43.21.18
nat (inside) 1 access-list 101
global  (outside) 1 209.165.201.10-209.165.201.20
```

А вот как это выглядит в новой версии:

```
object network myLAN
   subnet 10.1.2.0 255.255.255.0
object network InetServer
   host 150.43.21.18
object network NAT-pool
   range 209.165.201.10 209.165.201.20
nat (inside,outside) source dynamic myLAN NAT-pool destination static
InetServer InetServer
```

Заметьте, что команда nat указывается в режиме глобальной конфигурации, а не является частью object network. И когда я указывал адрес назначения, то не случайно указал два раза подряд сетевой объект InetServer. После указания ключевых слов destination static мы можем указать в команде nat сначала транслируемый IP-адрес, а затем реальный IP-адрес назначения. Это могло бы нам пригодиться, если бы мы хотели в одном правиле NAT транслировать и адрес источника и адрес назначения. Однако в большинстве сценариев нам это не потребуется, т.к. чаще всего адрес назначения будет общедоступным IP-адресом в Интернет. Оставить адрес назначения без изменения можно при помощи Identity NAT, указывая один и тот же адрес (в нашем случае это 150.43.21.18) и в качестве транслируемого и в качестве реального. Таким образом, трансляция адреса происходит в самого себя и получается, что трансляции как будто нет.

Вообще, указывать адрес назначения необязательно, несмотря на то, что это является основной отличительной особенностью Twice NAT. Но если мы его указываем, то необходимо сконфигурировать статическую трансляцию для этого адреса или просто использовать Identity NAT.

**Dynamic Twice NAT with Different Destination Ports**

В Twice NAT мы также как и в Network Object NAT можем транслировать не только адреса, но и порты. Представим, что наши клиенты локальной сети обращаются к одному и тому же адресу в

Интернет, но используют разные порты назначения. Например, если хосты из локальной сети обращаются к серверу 209.165.202.11 в Интернет по порту 80, то трансляция должна происходить в  адрес 209.165.201.129. Если же обращение идет к этому же серверу, но уже по порту 23, то трансляция должна происходить в адрес 209.165.201.130 (Рисунок 6).

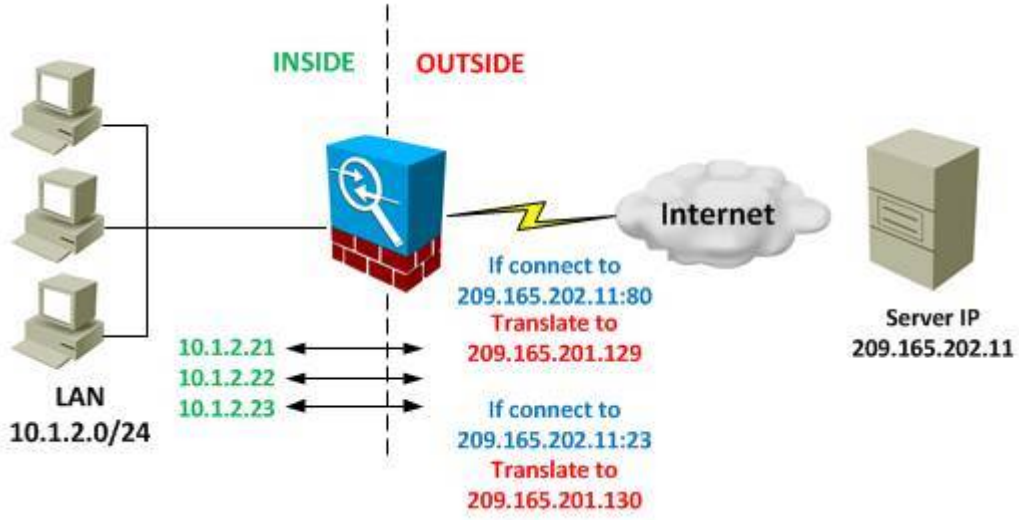


Рисунок 6

Так мы бы решили эту задачу раньше, до версии 8.3:

```
access-list WEB extended permit tcp 10.1.2.0 255.255.255.0 host
209.165.202.11 eq 80
access-list TELNET extended permit tcp 10.1.2.0 255.255.255.0 host
209.165.202.11 eq 23
nat (inside) 10 access-list WEB
global (outside) 10 209.165.201.129
nat (inside) 11 access-list TELNET
global (outside) 11 209.165.201.130
```

В новой версии это выглядит следующим образом:

```
object network myLAN
   subnet 10.1.2.0 255.255.255.0
object network Server-IP
   host 209.165.202.11
object network PAT-IP1
   host 209.165.201.129
object network PAT-IP2
   host 209.165.201.130
object service HTTP-SVC
   service tcp destination eq http
object service Telnet-SVC
   service tcp destination eq telnet
nat (inside,outside) source dynamic myLAN PAT-IP1 destination static
Server-IP Server-IP service HTTP-SVC HTTP-SVC
nat (inside,outside) source dynamic myLAN PAT-IP2 destination static
Server-IP Server-IP service Telnet-SVC Telnet-SVC
```

В этом примере появился новый тип объектов – сервисные объекты (**object service**). Они часто используются в Twice NAT и могут описывать как порты источника, так и порты назначения.  NAT поддерживает только протоколы TCP и UDP. Причем в одном правиле NAT используется какой-то

один из них.  Т.к. в нашем примере мы не транслируем адрес и порт назначения, то снова используем Identity NAT и для IP адреса и для номера порта.

Чтобы лучше понять, как в Twice NAT работает трансляция адреса назначения, давайте рассмотрим еще один пример.

**Static Twice NAT**

Представим, что нам необходимо транслировать в IP-адрес 209.165.200.228 трафик, который идет от реального IP 10.1.2.5 к отображенному IP 10.1.2.100, а также трафик, идущий от реального IP 209.165.200.225 к отображенному IP 209.165.200.228, транслировать в IP-адрес 10.1.2.100 (Рисунок 7).

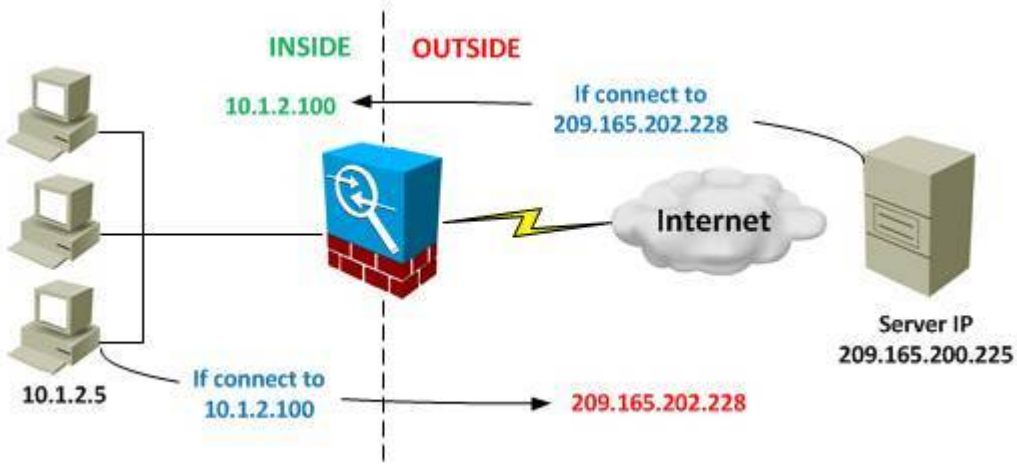


Рисунок 7

Вот как это настраивалось до версии 8.3:

```
access-list NET1 permit ip host 10.1.2.5 host 10.1.2.100
access-list NET2 permit ip host 209.165.200.225 host 209.165.200.228
static (inside,outside) 209.165.200.228 access-list NET1
static (outside,inside) 10.1.2.100 access-list NET2
```

А вот так это выглядит в версии 8.3:

```
object network myLANhost
   host 10.1.2.5
object network PAT-IP
   host 209.165.200.228
object network Server-IP
   host 209.165.200.225
object network Server-MappedIP
   host 10.1.2.100
nat (inside,outside) source static myLANhost PAT-IP destination static
Server-MappedIP Server-IP
```

**Policy NAT Exemption**

Для построения VPN-туннелей раньше мы использовали команду nat 0, но в новых версиях ОС для ASA её больше нет. Как же быть? Ответ прост – Twice NAT!

Раньше для построения VPN-туннеля мы с помощью ACL отфильтровывали трафик, идущий из одной сети в другую, и исключали его из трансляции (Рисунок 8 ).

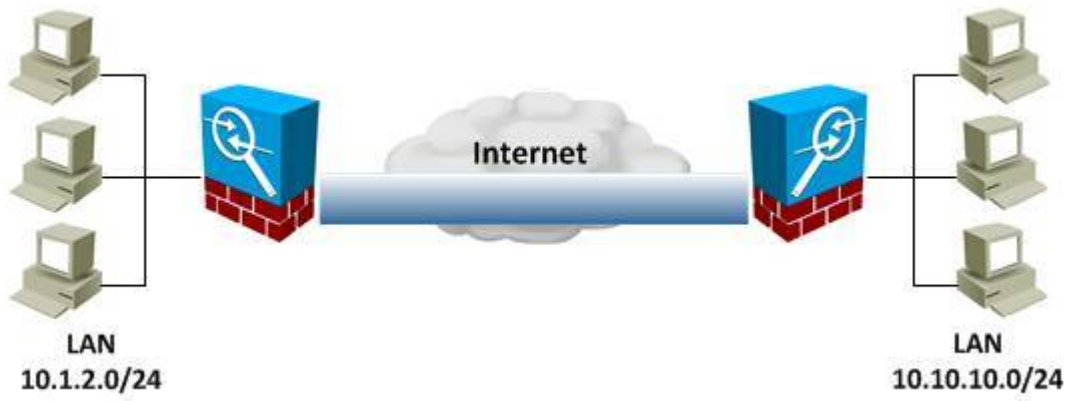


Рисунок 8

Выглядело это так:

```
access-list VPN-NO-NAT permit ip 10.1.2.0 255.255.255.0 10.10.10.0
255.255.255.0
nat (inside) 0 access-list VPN-NO-NAT
```

А вот как это выглядит сейчас:

```
object network myLAN
   subnet 10.1.2.0 255.255.255.0
   nat (inside,outside) dynamic interface
```

Это мы заодно выпустили клиентов локальной сети в Интернет

```
object network VPN-subnet
   subnet 10.10.10.0 255.255.255.0
nat (inside,outside) source static myLAN myLAN destination static VPN-
subnet VPN-subnet
```

Таким образом, мы использовали в правиле Twice NAT созданный ранее сетевой объект Network Object NAT, который уже описывает трансляцию адресов клиентов локальной сети для выхода в интернет. Вообще повторное использование объектов очень удобно и помогает нам при изменении каких-либо условий не редактировать множество строк кода, а внести изменения только в один или несколько объектов. Поэтому грамотно продуманная структура объектов поможет в дальнейшем упростить администрирование конфигурации ASA и снизить объем этой конфигурации.

Хорошо, мы построили VPN-туннель между сетями. А если нам необходимо принять соединение удаленных VPN-клиентов? Это не трудно сделать. Выделим им пул адресов (172.16.101.0/24) и создадим правило NAT:

```
object network Remote-VPN-pool
   subnet 172.16.101.0 255.255.255.0
nat (outside,inside) source static Remote-VPN-pool Remote-VPN-pool
destination static myLAN myLAN
```

Теперь клиенты удаленного доступа смогут получить доступ к ресурсам нашей локальной сети.

**NAT Control**

Начиная с версии 8.3 команда nat-control является устаревшей и больше не используется. Рекомендуется вместо этой команды использовать ACL для контроля трафика между интерфейсами с разным уровнем безопасности. При миграции на версию 8.3 вместо команды nat-control в конфигурационном файле вы сможете увидеть несколько правил Network Object NAT (в зависимости от количества интерфейсов на вашей ASA), которые будут любой трафик, идущий с интерфейса с большим уровнем безопасности в интерфейс с меньшим уровнем безопасности подвергать трансляции в адрес 0.0.0.0/32. Поэтому, готовясь к переходу на новую версию, не забудьте про этот момент.

**Порядок обработки правил NAT**

Мы рассмотрели с вами только основные изменения, произошедшие в трансляции адресов с появлением ОС версии 8.3. Еще нам осталось рассмотреть порядок обработки новых правил.

Новый порядок правил NAT использует таблицу из трех секций. В каждую из секции правила помещаются в зависимости от типа NAT. Обработка правил идет сверху вниз до первого совпадения, как в ACL.

В первую секцию попадают правила **Twice NAT**. Изначально их порядок следования в этой секции зависит от последовательности ввода этих правил в конфигурацию. Но если вам необходимо, чтобы одно из правил следовало раньше или позже других, то в правиле Twice NAT есть параметр line, с помощью которого можно указать номер строки, в которой должно размещаться правило.

Увидеть существующий порядок следования правил NAT на интерфейсе можно с помощью команды:

```
show nat interface <nameif>
```

Удалить ненужное правило NAT:

```
no nat <number of line>
```

Во вторую секцию попадают все правила **Network Object NAT**. Эти правила размещаются в секции 2 в автоматическом порядке, но всё же подчиняются некоторым правилам, которые надо знать.

Первыми в этой секции следуют правила Static NAT, после них идут правила Dynamic NAT. Правила одного типа также выстраиваются в определенном порядке:

1. Сначала идут правила транслирующие наименьшее количество адресов. Например, правило трансляции одного хоста будет выше, чем правило трансляции диапазона из четырех адресов или сети /24.
2. Правила, использующие младшие адреса IP-адреса, будут располагаться выше правил, использующих IP-адреса старшего порядка. Например, правило с IP-адресами сети 10.1.1.0/24 будет находиться в таблице выше, чем такое же правило с адресами сети 10.1.2.0/24.
3. Если в правилах используются одни и те же IP-адреса, то порядок следования правил будет зависеть от имён объектов, которые описывают правило трансляции. Они будут следовать в алфавитном порядке. Т.е. object network **LAN** будет находиться в таблице правил выше, чем такой же объект, но с названием **MyLAN**.

В третью секцию попадают правила **Twice NAT**, которые мы поместим туда сами с помощью специального параметра команды NAT, который называется after-auto. Это делается в случае, если мы хотим поместить какие-либо правила **Twice NAT** после правил **Network Object NAT**. Порядок правил в этой секции так же зависит от последовательности ввода этих правил в конфигурацию. Этот порядок мы также можем менять вручную с помощью параметра line команды Twice NAT.

**Заключение**

В этой статье мы рассмотрели основные изменения команд NAT, которые появились с выходом новой версии ОС Cisco ASA 8.3. На мой взгляд, правила NAT стали выглядеть «читабельнее» и интуитивно понятнее. Кажущееся увеличение объема конфигурации на деле оборачивается экономией времени, которое требуется, чтобы внести изменения в существующую конфигурацию. Это далеко не все изменения. В одной из следующих статей я рассмотрю изменения, касающиеся ACL. Также хочу заметить, что с момента выхода ОС версии 8.3(1) появились более свежие версии этой ОС, такие как 8.3(2), 8.4(1) и 8.4(2). В этих версиях произошли некоторые изменения в командах, относящихся к Identity NAT. Но основная концепция, изложенная в этой статье, остаётся неизменной. Появляется новый функционал, исправляются ошибки в работе старого функционала. В общем, работа идет, прогресс налицо. А нам надо стараться не отставать от этого прогресса и постоянно совершенствовать свои знания.