SRX SERIES AND J SERIES NETWORK ADDRESS TRANSLATION

Configuring Next-Generation NAT on Juniper Networks SRX Series Services Gateways and J Series Services Routers

Copyright © 2010, Juniper Networks, Inc.

Table of Contents

Copyright © 2010, Juniper Networks, Inc.

Table of Figures

Introduction

With the introduction of Juniper Networks® SRX Series Services Gateways, the Network Address Translation (NAT) feature in Junos® operating system release 9.2 has been restructured and separated from security policy. The new NAT architecture provides increased deployment flexibility and ease of use improvements. After reading this application note, users will be able to easily configure and troubleshoot NAT on SRX Series Services Gateways and J Series Services Routers.

Scope

The purpose of this application note is to explain Juniper Networks SRX Series Services Gateways' NAT architecture, and to provide several common configuration examples. This paper assumes that the reader is familiar with NAT and how it is used in both service provider and enterprise networks.

This new NAT architecture will also be migrated to Juniper Networks J Series Services Routers in Juniper Networks Junos OS release 9.5.

Design Considerations

Hardware Requirements

- Juniper Networks SRX Series Services Gateways
- Juniper Networks J2320, J2350, J4350, and J6350 Services Routers

Software Requirements

- Junos OS release 9.2 or later for all SRX Series Services Gateways
- Junos OS release 9.5 or later for all Juniper Networks J Series Services Routers

Description and Deployment Scenario

Network address translation is a technique that allows the mapping of private and public IP addresses. Its widespread use has, to a large extent, been ensured by the exhaustion of public IPs and the need to conserve those that remain. Junos OS has supported several NAT methods since its introduction.

The current J Series flow-based NAT architecture is largely patterned after ScreenOS® NAT, where security policies are used to specify which packets require address translation. This implementation, however, has some limitations (mostly on the usability side), and these factors have driven the new SRX Series and J Series design.

Original Design and Processing

In the original model for J Series Services Routers, packets were processed according to the following simplified diagram:

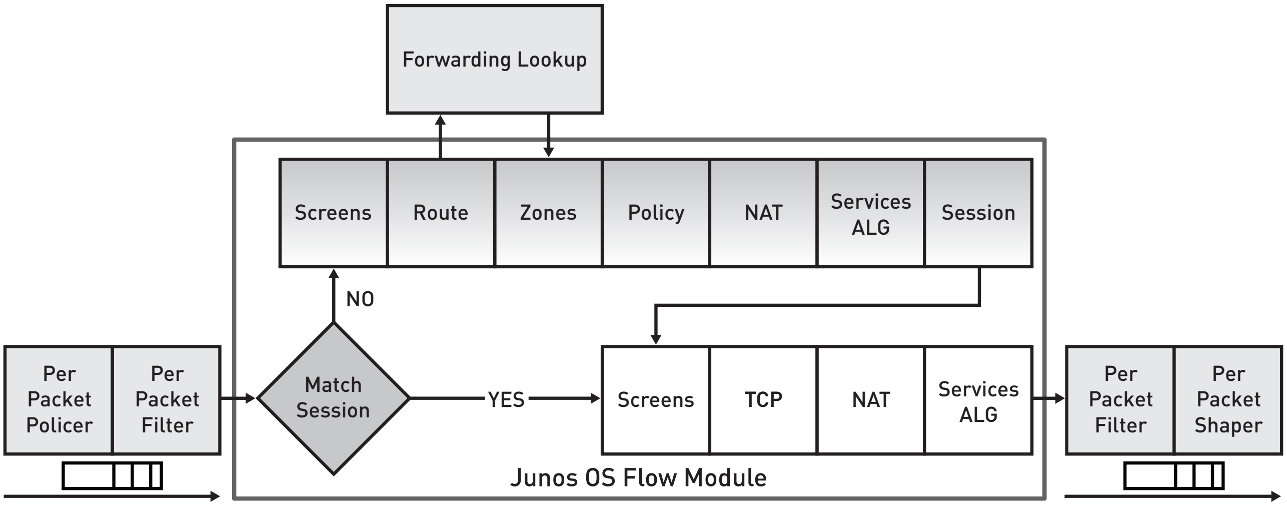


Figure 1: Packet flow

Copyright © 2010, Juniper Networks, Inc.

As shown, security policies control how packets are processed, which artificially limits NAT functionality and creates unnecessary interdependencies. For example:

- NAT rules and security policies are indivisible. In order to translate the address of a packet (or more generally, all packets in a session), a policy must be created that matches packets and specifies NAT as one of the actions.
- Because policies are used to control both NAT and the traffic allowed through a device, changes in the NAT configuration necessitate changes in security policies and vice-versa.
- Since security policies use zones to classify traffic (when a packet from a new session is processed, the source/destination zone combination is used to determine which security policies are matched against the packet), some NAT configurations require multiple policies.

  For example, imagine a device where the Internet is accessed through a private network (where NAT is eventually applied downstream), or is accessed directly. From a security point of view, the requirements are identical, regardless of the egress interface the packets take. The NAT configuration for this case requires users to duplicate security policies even when the security posture is the same. Policies matching traffic destined to the private network would have no NAT configuration, while the ones matching traffic routed to the Internet would.
- On some occasions (most notably when doing destination NAT), the destination IP address of received packets is not the address of a network connected to the device, and therefore may not be in the routing table (this destination address is meant to be translated to the address of the real destination hosts). In this situation, the policy lookup fails resulting in packet discards. To circumvent this limitation, users must create “fake” routes pointing to the destination address of the packets (even when these addresses are going to be translated), which allows the policy lookup for those packets. If NAT were processed first, then the fake routes would not be needed because the policy lookup would take place using the translated address and not the original address of the packets.

Juniper’s new architecture decouples NAT processing from security policies, allowing NAT rules to be configured independently and removing the restrictions imposed by the old processing model.

New Design and Processing

As previously mentioned, next-generation NAT is independent of security policies, but remains part of the flowd module.

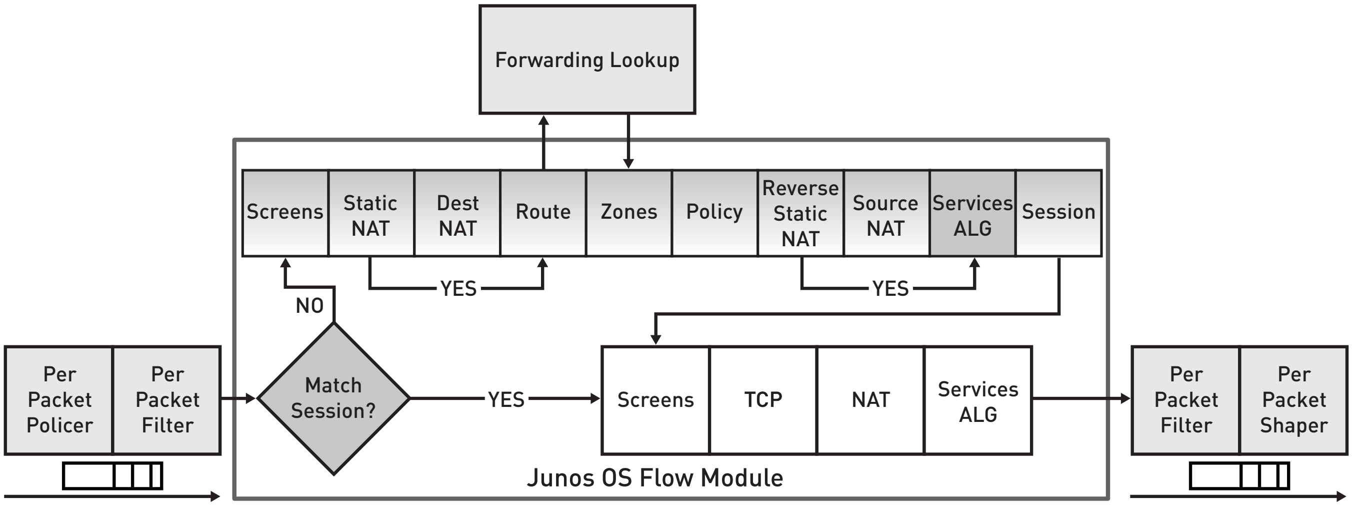


Figure 2: New packet flow

In particular,

- There are no changes to the fast-path. Once a session is established, packet processing remains the same. The change is limited to the slow-path used to establish the initial session.
- In the slow-path, NAT is distributed depending on the kind of translation to be done to accommodate the different use cases relevant to each NAT type. For instance, destination and static NAT are processed before route lookup and security policy processing, which eliminates the need to install fictitious routes in the routing table as described above.

Copyright © 2010, Juniper Networks, Inc.

NAT processing centers around the evaluation of NAT rule-sets. A NAT rule-set is a collection of rules, each of which determines what traffic is processed and what method is used. In a sense, NAT rule-sets are not much different than a security context (a source/destination zone combination). A rule-set determines the overall direction of the traffic to be processed. For example, a rule-set can select traffic from a particular interface or to a given zone, while each rule in the rule-set specifies the traffic that matches the rule (by matching on the source/destination address of the traffic) and the action to be taken. Rule-sets can contain multiple rules (up to 8 rules per rule-set in Junos OS 9.3, although this limit may change in future versions).

Each NAT rule-set specifies a set of matching conditions that include:

- Source/destination interface
- Source/destination zone
- Source/destination routing instance

The configuration model is the same for each type of NAT (source, destination, or static) with the exception that some NAT rule-sets can specify more matching conditions than others. Since both destination and static NAT are processed before route lookup, destination and static NAT rules cannot match on the destination zone, interface, or routing instance of the packets.

Once a matching rule-set has been found, each rule belonging to the rule-set is evaluated for a match. NAT rules can match on:

- Source/destination address
- Destination port (only for destination NAT as of Junos OS version 9.3)

Finally, once a packet matches a rule in a rule-set during session establishment, traffic is processed according to the action specified by that rule. The configuration hierarchy found under [security nat] for each NAT type is as follows:

Source NAT Configuration Hierarchy

```
source {
        address-persistent
        pool {…}
        pool-utilization-alarm {…}
        rule-set <rule-set name>{
            from {
                interface <interface list>;
                zone <zone list>;
                routing-instace <routing-instance list>;
            }
            to {
                interface <interface list>;
                zone <zone list>;
                routing-instace <routing-instance list>;
            }
            rule <rule-name> {
                match {
                    source-address <source address/prefix list>;
                    destination-address <source address/prefix list>;
                }
                then source-nat {
                    inerface | off | pool <pool-name>;
                }
            }
        }
}
```

Copyright © 2010, Juniper Networks, Inc.

Destination NAT Configuration Hierarchy

```
destination {
        pool {…}
        rule-set <rule-set name>{
            from {
                interface <interface list>;
                zone <zone list>;
                routing-instace <routing-instance list>;
            }
            rule <rule-name> {
                match {
                    source-address <source address/prefix list>;
                    destination-address <source address/prefix list>;
                    destination-port <destination port>;
                }
                then destination-nat {
                    off | pool <pool-name>;
                }
            }
        }
}
```

Static NAT Configuration Hierarchy

```
static {
        rule-set <rule-set name>{
            from {
                interface <interface list>;
                zone <zone list>;
                routing-instace <routing-instance list>;
            }
            rule <rule-name> {
                match {
                    destination-address <source address/prefix list>;
                }
                then static-nat {
                    prefix <address prefix>;
                    routing-instance <instance-name>;
                }
            }
        }
}
```

NAT pools are used to specify the address and some of the translation parameters. Since pools can be used both for source and destination NAT, the configurable options depend on the type of NAT required.

Source NAT pools specify:

- The address, which can be a single address, a prefix, or an address range
- The ports, in the form of a range when port translation is used, or no-translation when port translation is not required (see the configuration examples for more details)
- The host base address used for address shifting
- The routing instance to which the pool belongs; default is the main routing instance
- Optionally, an overflow pool should the pool run out of addresses, which can only be used for pools with no-port-translation

Copyright © 2010, Juniper Networks, Inc.

Destination NAT pools specify:

- The destination address or address range
- The destination port, which is used for port forwarding, as shown in the example associated with Figure 11
- The routing instance to which the pool belongs; the default is the main routing instance.

During processing, it is possible for a packet to match more than one rule-set. To sort out this ambiguity, Junos OS always selects the rule-set with a more specific match. In particular, an interface match is considered more specific than a zone match, which in turn is more specific than a routing-instance match.

Rules inside a rule-set are evaluated in order, where the first rule matching the packet is the one used for processing. The evaluation of rule-sets and rules can be represented as follows:

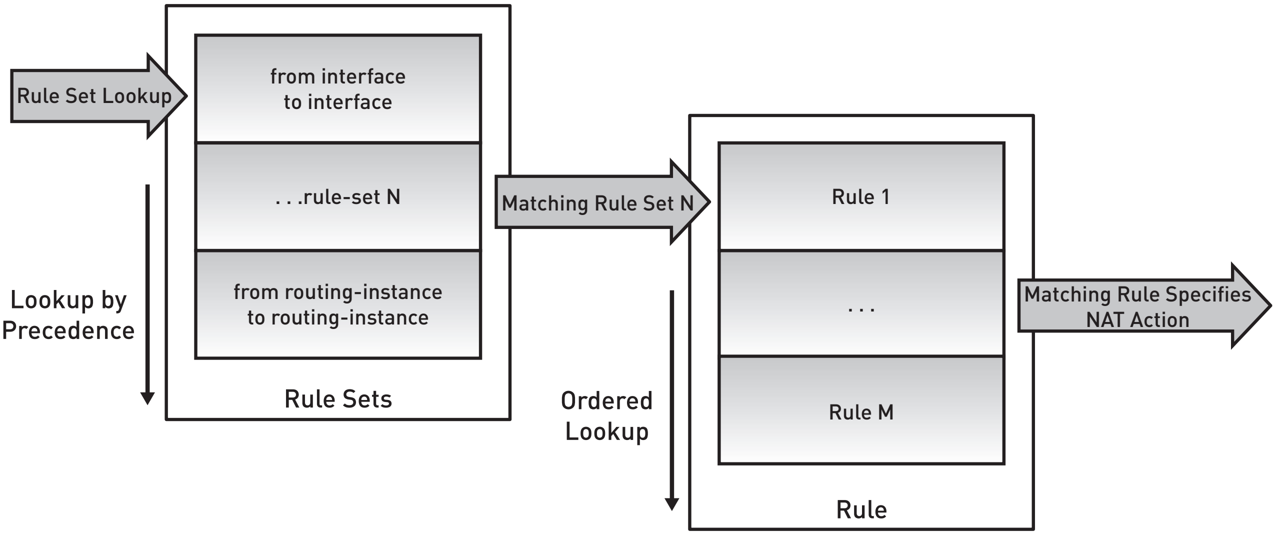


Figure 3: NAT rule-set evaluation priority

Source NAT

Source NAT is frequently used to perform the following common actions:

- Translate a single IP address to another address (used to provide a single device behind a private network access to the Internet)
- Translate a contiguous block of addresses to another block of the same size
- Translate a contiguous block of addresses to another block of smaller size
- Translate a contiguous block of addresses to a single IP or smaller block using port translation.
- Translate a contiguous block of addresses to the address of the egress interface

One-to-one and many-to-many translations for address blocks of the same size do not require port translation, since for every address there is an available address in the pool. When the size of the address pools is smaller than the number of addresses to be translated, either the total number of concurrent addresses to be translated is limited by the size of the pool (for instance, if a full class C with up to 253 devices is translated to a pool of size 10, only up to 10 devices can connect simultaneously), or port translation has to be used.

Source NAT actions 4 and 5 are identical with the exception that the address of the egress interface is used for the translation in action #5. Because the translation pool contains only one IP (the interface's address), interface translation always uses port translation. Scenario 5 does not make use of an address pool (the configuration for this scenario is shown in Figure 8).

The configuration syntax for Source NAT actions is shown in Table 1.

Table 1: Source NAT Pool Configurations

Scenario	Configuration

Single IP to single IP (case 1)	pool <pool name> {
	address <IP address>/32;
	}

Many-to-many with blocks of the same size (case 2)	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	host-address-base <ip address>
	}

Many-to-many with blocks of different sizes and no port translation (case 3)	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	port no-translation;
	}

Many-to-many with blocks of different sizes and port translation (case 4)	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	}

Destination NAT

Just like in the source NAT case, the address pool determines how many addresses are available for translation. The main difference between these two types of translation is that destination NAT always uses a 1-to-1 correspondence between the translated and un-translated addresses. Destination NAT is frequently used to perform the following common actions:

1. Translate a single IP address to another address, which is often used to allow devices on the Internet to connect to a host in a private network.
2. Translate a contiguous block of addresses to another block of the same size, which could be used to allow access to a group of servers.
3. Translate a destination IP/port combination to another destination IP/port, which may be used to allow access to multiple services using the same IP address but different ports, as shown in Figure 11.

The configuration syntax for Destination NAT actions is shown in Table 2.

Table 2: Source NAT Pool Configurations

Scenario	Configuration

Single IP to single IP (case 1)	pool <pool name> {
	address <IP address>/32;
	}

Many-to-many with blocks of the same size (case 2)	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	}

Specific IP/port combination to another IP/port	pool <pool name> {
	address <IP address>/32;
	port <port number>;
	}

Copyright © 2010, Juniper Networks, Inc.

Static NAT

In the previous cases, both source and destination NAT allow connections to be initiated ONLY from one side of the network. Source NAT facilitates outgoing connections (for example, from the private network to the Internet), while destination NAT facilitates incoming connections only (for example, from the Internet to a server on a private network).

Static NAT allows connections to be originated from either side of the network, but is limited in that only one-to-one translations are possible (or many-to-many with blocks of the same size). In other words, in order to allow connections to be initiated from either side, a public address MUST be allocated for each private address.

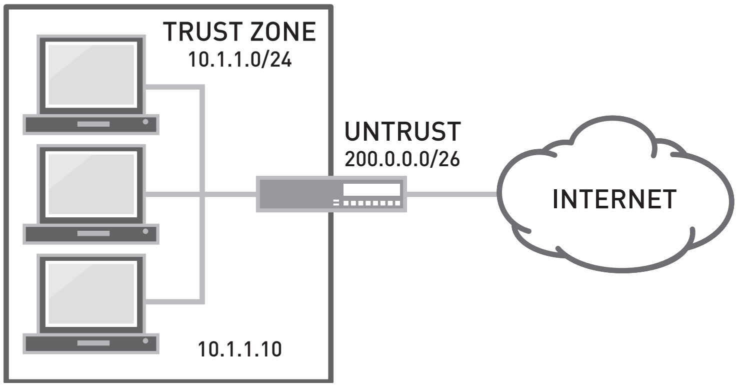
Since the deployment combinations are much simpler, no address pools are needed. For each matching rule, a single prefix has to be used for the translation, and both the matching prefix and the translation prefix must be the same size.

Configuration Examples

The following section provides several configuration examples to illustrate the different deployment scenarios previously discussed. Most of the examples follow the same structure. Each example includes an example network design, the translated and un-translated addresses, and the required configuration.

Source NAT Configuration Examples

Source NAT—Single Address Translation

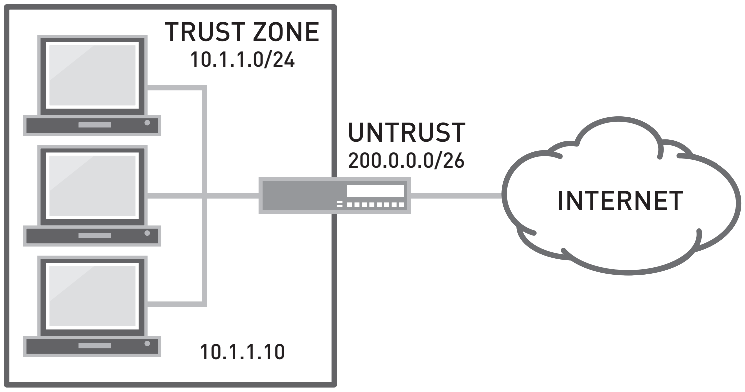


Original Source	Translated Source
10.1.1.10	200.0.0.10

Figure 4: Single address translation

```
source {
    pool 200_0_0_10 {
        address {
            200.0.0.10/32;
        }
    }
    rule-set one-to-one {
        from zone trust;
        to zone untrust;
        rule single-ip-nat {
            match {
                source-address 10.1.1.10/32;
            }
            then {
                source-nat pool 200_0_0_10;
            }
        }
    }
}
```

Source NAT—Many-to-Many Translation with Address Blocks of the Same Size



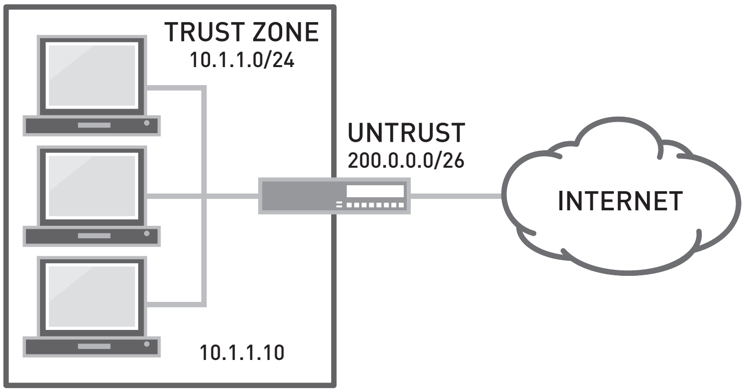
Original Source	Translated Source
10.1.1.10 to 10.1.1.20	200.0.0.30 to 200.0.0.40

Figure 5: Address shifting

```
source {
    pool address-shifting {
        address {
            200.0.0.30/32 to 200.0.0.40/32;
        }
        host-address-base 10.1.1.10/32;
    }
    rule-set address-shift {
        from zone trust;
        to zone untrust;
        rule net-10_1_1_0 {
            match {
                source-address 10.1.1.0/24;
            }
            then {
                source-nat pool address-shifting;
            }
        }
    }
}
```

This example deserves some discussion. The host-address-base option specifies which of the original source addresses starts the address range. This keyword is required because the matching conditions of the rule-set do not allow the specification of an address range (only prefixes). Packets with a source address that match the rule source-address, but are not in the [host-address-base, host-address-base+pool size] range, are not translated.

Source NAT—Many-to-Many Translation with Address Blocks of Different Size



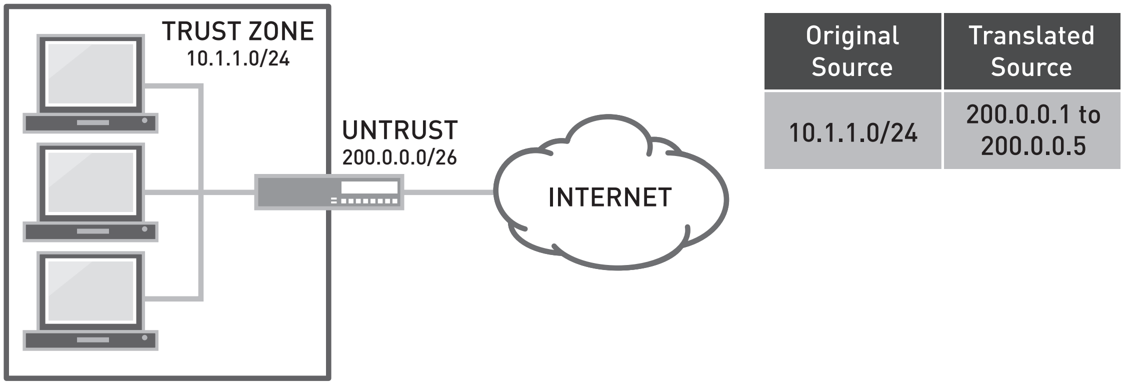
Original Source	Translated Source
10.1.1.0/24	200.0.0.30 to 200.0.0.40

Figure 6: Many-to-many with no port translation

Copyright © 2010, Juniper Networks, Inc.

```
source {
    pool many-no-port-translation {
        address {
            200.0.0.30/32 to 200.0.0.40/32;
        }
        port no-translation;
    }
    rule-set address-shift {
        from zone trust;
        to zone untrust;
        rule net-10_1_1_0 {
            match {
                source-address 10.1.1.0/24;
            }
            then {
                source-nat pool many-no-port-translation;
            }
        }
    }
}
```

Source NAT—Many-to-Many Translation with Address Blocks of Different Size and Port Translation



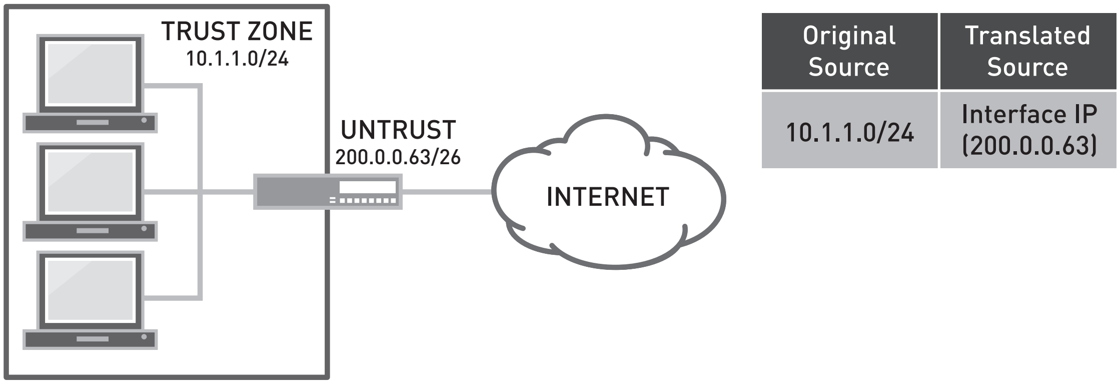
Original Source	Translated Source
10.1.1.0/24	200.0.0.1 to 200.0.0.5

Figure 7: Many-to-many with port translation

```
source {
    pool many-to-many {
        address {
            200.0.0.30/32 to 200.0.0.40/32;
        }
    }
    rule-set address-shift {
        from zone trust;
        to zone untrust;
        rule net-10_1_1_0 {
            match {
                source-address 10.1.1.0/24;
            }
            then {
                source-nat pool many-to-many;
            }
        }
    }
}
```

Copyright © 2010, Juniper Networks, Inc.

Source NAT—Interface Translation



Original Source	Translated Source
10.1.1.0/24	Interface IP (200.0.0.63)

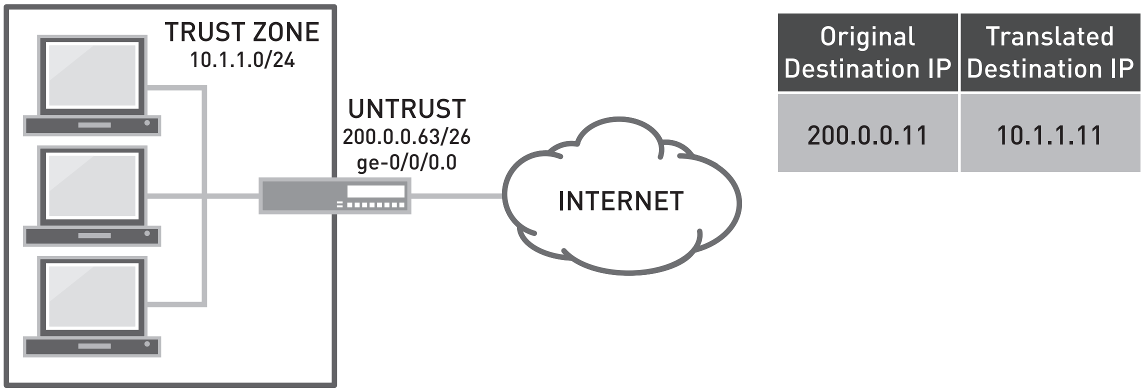
Figure 8: Interface NAT

```
source {
    rule-set interface-nat {
        from zone trust;
        to zone untrust;
        rule net-10_1_1_0 {
            match {
                source-address 10.1.1.0/24;
            }
            then {
                source-nat interface;
            }
        }
    }
}
```

The use of the “source-nat interface” keyword provides dual functionality. It allows packets to be translated to the address of the egress interface. It also allows for NAT even when the address of the egress interface is unknown—for example, when the address is dynamically assigned by Dynamic Host Configuration Protocol (DHCP) or Point-to-Point Protocol (PPP).

Destination NAT Configuration Examples

Destination NAT—Single Address Translation



Original Destination IP	Translated Destination IP
200.0.0.11	10.1.1.11

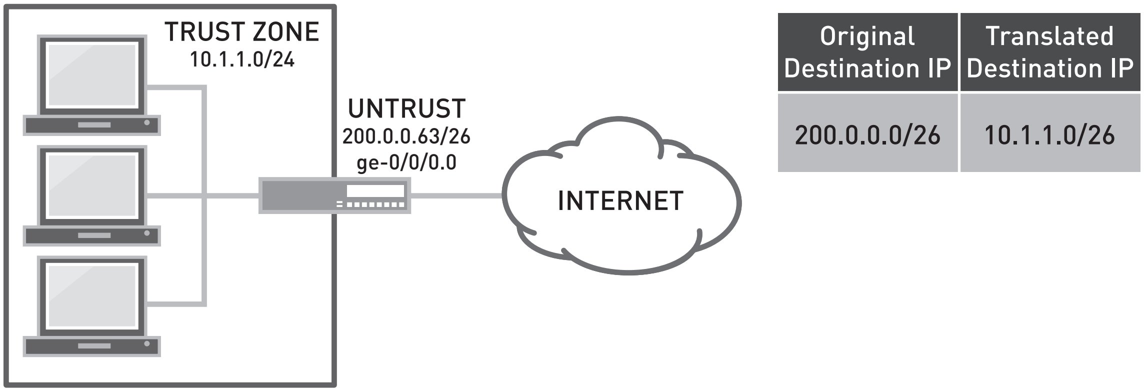
Figure 9: Single address destination translation

Copyright © 2010, Juniper Networks, Inc.

```
destination {
    pool server-1 {
        address 10.1.1.11/32;
    }
    rule-set nat-example {
        from interface ge-0/0/0.0;
        rule single-address-nat {
            match {
                destination-address 200.0.0.11/32;
            }
            then {
                destination-nat pool server-1;
            }
        }
    }
}
proxy-arp {
    interface ge-0/0/0.0 {
        address {
            200.0.0.11/32;
        }
    }
}
```

In this example, the address 200.0.0.11 belongs to the Untrust interface ge-0/0/0.0 subnet. In order for this interface to respond to Address Resolution Protocol (ARP) requests to the 200.0.0.11 address, (the public address translated to 10.1.1.11) proxy-arp has to be configured on the interface as shown below.

Destination NAT—Many-to-Many Translation



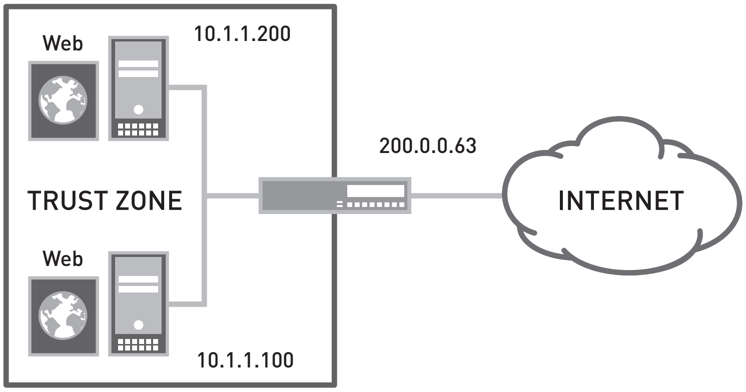
Original Destination IP	Translated Destination IP
200.0.0.0/26	10.1.1.0/26

Figure 10: Many-to-many destination translation

```
destination {
    pool trust-net {
        address 10.1.1.0/26;
    }
    rule-set nat-example {
        from interface ge-0/0/0.0;
        rule many-to-many-translation {
            match {
                destination-address 200.0.0.0/16;
            }
            then {
                destination-nat pool trust-net;
            }
        }
    }
}
proxy-arp {
    interface ge-0/0/0.0 {
        address {
            200.0.0.1/32 to 200.0.0.62/32;
        }
    }
}
```

Like in the previous example, the NAT pool belongs to the address range of the ge-0/0/0.0 interface, so proxy ARP is enabled for all addresses in the pool, with the exception of the IP address of the interface itself.

Destination NAT—IP/Port Translation



Original Destination IP	Translated Destination IP
200.0.0.63 port 80	10.1.1.100 port 80
200.0.0.63 port 8080	10.1.1.200 port 80

Figure 11: Port forwarding

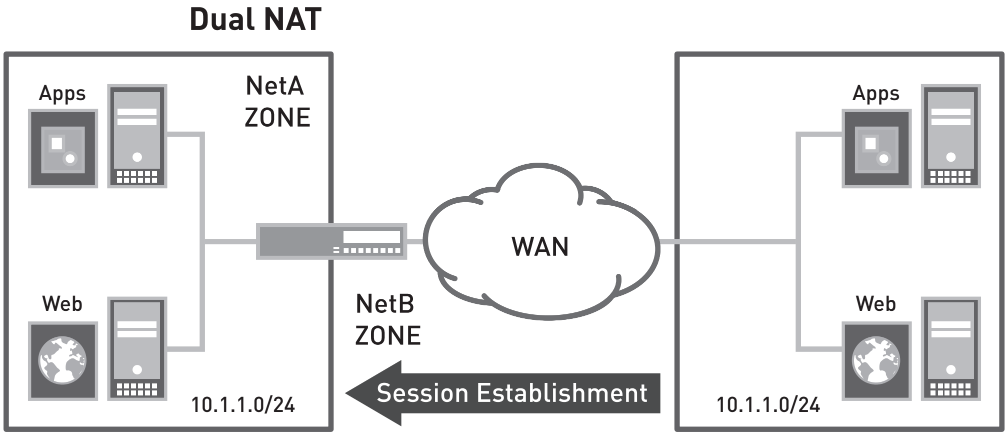
Copyright © 2010, Juniper Networks, Inc.

```
destination {
    pool server-1 {
        address 10.1.1.100/32 port 80;
    }
    pool server-2 {
        address 10.1.1.200/32 port 80;
    }
    rule-set nat-example {
        from interface ge-0/0/0.0;
        rule port-forwarding {
            match {
                destination-address 200.0.0.63/32;
                destination-port 80;
            }
            then {
                destination-nat pool server-1;
            }
        }
        rule port-forwarding-2 {
            match {
                destination-address 200.0.0.63/32;
                destination-port 8080;
            }
            then {
                destination-nat pool server-2;
            }
        }
    }
}
```

Static NAT Configuration Examples

In this example, we will redo the configuration introduced in Figure 10, but use static NAT as an alternative. The only difference in this example is that the new configuration will allow connections to be established from either side of the gateway, where Destination NAT—Many-to-Many Translation only allowed connections established from the Internet to the 10.1.1.0/26 network).

```
static {
    rule-set nat-example {
        from interface ge-0/0/0.0;
        rule nat-trust-net {
            match {
                destination-address 200.0.0.0/26;
            }
            then {
                static-nat prefix 10.1.1.0/26;
            }
        }
    }
}
proxy-arp {
    interface ge-0/0/0.0 {
        address {
            200.0.0.1/32 to 200.0.0.62/32;
        }
    }
}
```



Original Source IP	Translated Source IP
10.1.1.0/24	10.2.0.0/24
Original Dest IP	Translated Dest IP
10.2.0.0/24	10.1.1.0/24

Figure 12: Dual NAT

This example involves the simultaneous translation of both source and destination addresses and is typically found on networks with address-space collision. When a host on NetB wants to connect to a host on NetA, it does so by sending a packet to the 10.2.0.0/24 network. For instance, if host 10.1.1.1 on NetB wants to establish a session to host 10.1.1.5 on NetA, it would send a packet to the 10.2.0.5 address. Packets received for the 10.2.0.0/24 network coming from the 10.1.1.0/24 network will be translated as going to the 10.1.1.0/24 network and coming from the 10.2.0.0/24 network, which allows the hosts on both ends to communicate using the 10.2.0.0/24 network as an intermediary.

```
source {
    pool intermediate-net {
        address {
            10.2.0.0/24;
        }
        port no-translation;
    }
    rule-set nat-example {
        from zone NetB;
        to zone NetA;
        rule double-nat-source {
            match {
                source-address 10.1.1.0/24;
            }
            then {
                source-nat pool intermediate-net;
            }
        }
    }
}
destination {
    pool trust-net {
        address 10.1.1.0/24;
    }
    rule-set nat-example {
        from zone NetB;
        rule double-nat-dest {
            match {
                destination-address 10.2.0.0/24;
            }
            then {
                destination-nat pool trust-net;
            }
        }
    }
}
```

Copyright © 2010, Juniper Networks, Inc.

NAT and Security Policies Interaction

Since NAT processing has been separated from security policies, both translated and un-translated addresses may match policies.

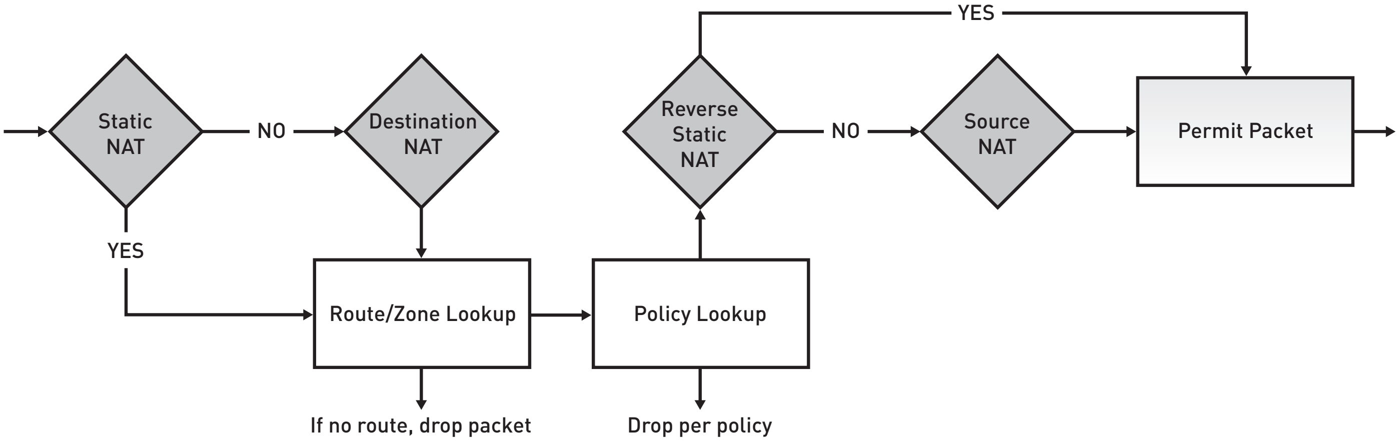


Figure 13: Policy-NAT processing

Figure 13 represents a simplified view of NAT processing (which can also be seen in Figure 2. New packet flow). By looking at the diagram, we see that:

- The policy engine will match translated addresses for Destination or Static NAT. In other words, packets will be translated before they enter the policy engine.
- The policy engine will match un-translated addresses for source or reverse Static NAT connections.

The design is such that firewall administrators only need to be concerned with the actual endpoints involved in the communication. A client on a private network establishing a connection to a server over the Internet (via source NAT) will require a policy matching the client private address and the server (public) address. A connection from an Internet host to a server in the private network (through a public IP using destination NAT) will again require a policy matching the client address and the server private (translated) address.

However, some ambiguity is possible. Assume that we have the following network:

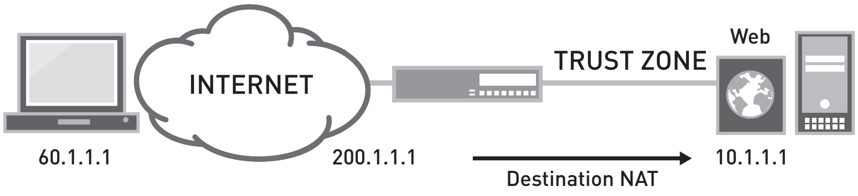


Figure 14: Dropping non-NAT traffic

In this network, the gateway performs destination NAT in order to allow Internet connections to the Web server in the Trust zone (with address 10.1.1.1). As already discussed, the policy engine will observe traffic from the client (60.1.1.1, in this example) to the translated address of server 10.1.1.1. The difficulty is that the policy engine will see traffic sent to the private address 10.1.1.1 as identical to traffic destined for the 200.1.1.1 address (which was translated to 10.1.1.1). In other words, traffic sent to the public address of the server (valid) and to the private address of the server (invalid) seem the same to the policy engine. To overcome this challenge, whenever destination or static NAT is used, the policy engine is informed that traffic has been translated, which allows policies to explicitly drop translated or un-translated traffic as shown in the example below.

```
from-zone untrust to-zone trust {
    policy reject-untranslated {
        match {
            source-address any;
            destination-address 10.1.1.1/32;
            application any;
        }
        then {
            permit {
                destination-address {
                    drop-untranslated;
                }
            }
        }
    }
}
```

Monitoring

Junos OS provides several commands used to verify, monitor, and troubleshoot NAT.

A good starting point is the session table, which lists all of the sessions currently tracked by the device. The session table includes the source/destination addresses of the packet, source/destination zones, source/destination ports, the protocol, and translation information if the traffic is translated.

The session table is displayed with the "show security flow session" command, which provides options to filter sessions based on the table data.

```
root@SRX210-1> show security flow session
…
Session ID: 3729, Policy name: nat-example-security-policy/6, Timeout: 2
  In: 10.1.0.13/52939 --> 207.17.137.229/80;tcp, If: fe-0/0/5.0
  Out: 207.17.137.229/80 --> 172.19.101.42/2132;tcp, If: ge-0/0/0.0
```

The above output shows traffic matching a policy named nat-example-security-policy. The "In" direction shows traffic as it was received by the gateway (that is, prior to any translation). The "Out" direction shows the traffic as it is expected to ingress (after it is translated). A connection from X to Y, which is translated to X' to Y', will show as X->Y for the "In" direction and Y'->X' for the "Out" direction. This allows an administrator to see not only all sessions, but also all translations.

In this example, we can see a TCP session established from the 10.1.0.13 client to a server with an address/port combination of 207.17.137.228 and 80 (which probably indicates HTTP traffic). The source address was translated from 10.1.0.13 port 52939 to 172.19.101.42 port 2132 (source NAT with port translation).

Another useful command is "show security nat source|destination|static rule all", which shows all configured NAT rules and the number of translation hits (the number of sessions that were translated by each rule).

Copyright © 2010, Juniper Networks, Inc.

```
run show security nat source rule all
Total rules: 1

source NAT rule: net-10_1_1_0         Rule-set: interface-nat
  Rule-Id    : 1
  From zone : trust
  To zone    : untrust
  Match
    Source addresses  :   10.1.1.0        - 10.1.1.255
    Action    : interface
    Translation hits  : 1112
```

The "show security nat interface-nat-ports" command displays all allocated ports being used for interface-based NAT. Finally, issuing "show security nat source|destination pool" shows configured pools, their size, and the number of translation hits.

```
show security nat source pool many-to-many

Pool name           : many-to-many
Pool id             : 4
Routing instance    : default
Host address base  : 0.0.0.0
Port                : [1024, 32255]
Total addresses     : 11
Translation hits    : 0
```

Summary

Beginning with Junos OS release 9.2, Juniper Networks has restructured NAT and separated it from security policy on SRX Series Services Gateways. The same improvements will be ported to J Series Services Routers in Junos OS release 9.5. This new architecture provides the flexibility to meet challenging network requirements while offering increased ease of use.

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at **www.juniper.net**.

Corporate and Sales Headquarters
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100
www.juniper.net

APAC Headquarters
Juniper Networks (Hong Kong)
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EMEA Headquarters
Juniper Networks Ireland
Airside Business Park
Swords, County Dublin, Ireland
Phone: 35.31.8903.600
EMEA Sales: 00800.4586.4737
Fax: 35.31.8903.601

To purchase Juniper Networks solutions, please contact your Juniper Networks representative at 1-866-298-6428 or authorized reseller.

Copyright 2010 Juniper Networks, Inc. All rights reserved. Juniper Networks, the Juniper Networks logo, Junos, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

3500151-002-EN May 2010

Printed on recycled paper