APPLICATION NOTE

SRXシリーズおよびJシリーズのネットワークアドレス変換

ジュニパーネットワークス SRX シリーズ サービス・ゲートウェイおよび J シリーズ サービスルーターにおける次世代 NAT の構成

目次

図目次

はじめに

ジュニパーネットワークス SRX シリーズ サービス・ゲートウェイの導入に伴い、Junos® OS リリース 9.2 の NAT（Network Address Translation）機能が再構築され、セキュリティポリシーから分離されました。新しい NAT アーキテクチャは、導入の柔軟性と使いやすさを向上させます。本書をお読みいただけば、SRX シリーズ サービス・ゲートウェイおよび J シリーズ サービスルーターにおいて、NAT の構成およびトラブルシューティングが簡単に行えるようになります。

本書の目的

本書では、ジュニパーネットワークス SRX シリーズ サービス・ゲートウェイの NAT アーキテクチャについて説明し、一般的な設定例をいくつか紹介します。本書は、NAT を熟知し、サービスプロバイダネットワークおよびエンタープライズ向けネットワークで NAT がどのように利用されるかをよく理解していることを前提としています。

この新しい NAT アーキテクチャは、ジュニパーネットワークス Junos OS リリース 9.5 により、ジュニパーネットワークス J シリーズ サービスルーターにも移植されます。

設計上の考慮事項

ハードウェア要件

- ジュニパーネットワークス SRX シリーズ サービス・ゲートウェイ
- ジュニパーネットワークス J2320、J2350、J4350、および J6350 サービスルーター

ソフトウェア要件

- Junos OS リリース 9.2 以上（SRX シリーズ サービス・ゲートウェイの場合）
- Junos OS リリース 9.5 以上（ジュニパーネットワークス J シリーズ サービスルーターの場合）

説明および導入シナリオ

ネットワークアドレス変換は、プライベート IP アドレスおよびパブリック IP アドレスのマッピングを可能にする技術です。パブリック IP が枯渇し、その節約の必要性が高まったことにより、この技術は幅広く普及しています。Junos OS は、導入当初からいくつかの NAT 方式をサポートしてきました。

現在の J シリーズで使用されているフローベースの NAT アーキテクチャは、大部分が ScreenOS® の NAT にならって作られており、セキュリティポリシーを使用してアドレス変換が必要なパケットが指定されます。しかし、このような実装には限界があったため（主に使い勝手の面で）、新しく SRX シリーズと J シリーズが設計されることになりました。

初期の設計および処理

J シリーズ サービスルーターの初期モデルでは、以下の簡易図の流れに従って、パケットが処理されていました。

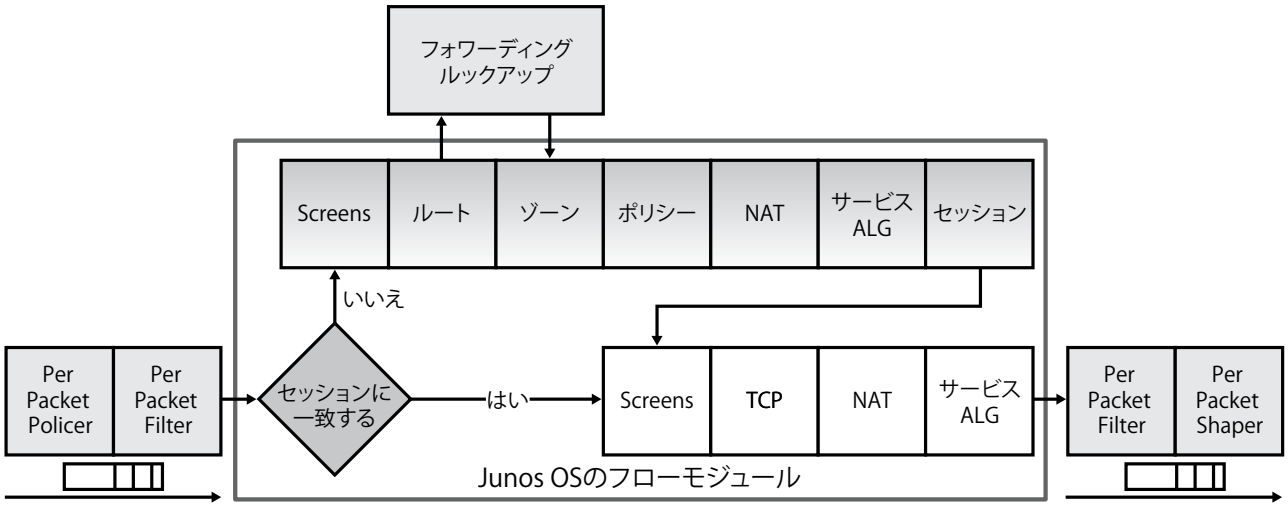


図1：パケットフロー

図に示されているとおり、パケットの処理方法がセキュリティポリシーによって制御されるため、NAT の機能に制約が生じ、不要な相互依存性が生じます。以下に例を示します。

- NAT ルールとセキュリティポリシーが不可分である。1つのパケットのアドレスを変換するには(セッション内のすべてのパケットを変換する方がより一般的)、パケットに一致し、アクションの1つとして NAT を指定するポリシーを作る必要がある。
- NAT とデバイスを通過したトラフィックの両方がポリシーで制御されるため、NAT 構成に変更があると、セキュリティポリシーも変更する必要がある（逆の場合も同じ)。
- セキュリティポリシーはゾーンを使ってトラフィックを分類するため（新しいセッションからのパケットを処理するとき、ソース / 宛先ゾーンの組み合わせからパケットに適合するセキュリティポリシーを判断する)、NAT 構成によっては複数のポリシーが必要になる場合がある。

  たとえば、私設網を介して（結局は下流で NAT が適用される)、または、直接インターネットにアクセスするデバイスがあるとする。セキュリティの観点からすると、パケットの出力インタフェースが何であっても要件は同じだが、この場合の NAT 構成では、セキュリティ条件が同じ場合でも、ユーザーはセキュリティポリシーを複製する必要がある。私設網宛てのトラフィックに一致するポリシーには NAT 構成がないのに、インターネットに送信されるトラフィックに一致するポリシーには NAT 構成があることになる。
- 場合によって（特に、宛先 NAT の場合)、受信したパケットの宛先 IP アドレスが、デバイスに接続されているネットワークのアドレスでないため、そのアドレスがルーティングテーブルに存在しないことがある（この宛先アドレスは、実際の宛先ホストのアドレスに変換される)。このような場合、ポリシー検索が失敗し、パケットが廃棄される。これを回避するには、パケットの宛先アドレスを指す「偽」ルートを作る必要がある(これらのアドレスが変換されることになっている場合でも)。それにより、これらのパケットのポリシー検索が可能になる。NAT が最初に処理されるのであれば、パケットの当初のアドレスではなく変換後のアドレスでポリシー検索が行われるため、偽ルートは必要なくなる。

ジュニパーネットワークスの新しいアーキテクチャは、セキュリティポリシーから NAT 処理を切り離すことにより、NAT ルールの独立した構成を可能にし、古い処理方式で課されていた制約を取り除きます。

新しい設計および処理

前述のとおり、次世代 NAT はセキュリティポリシーから独立していますが、フローモジュールの一部であることに変わりはありません。

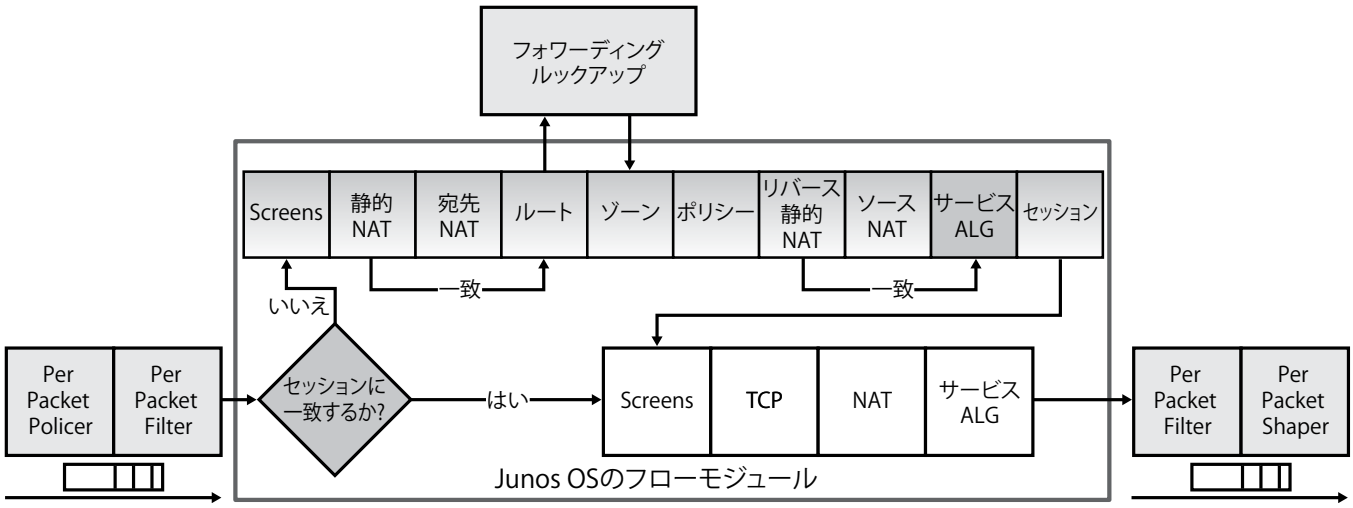


図 2：新しいパケットフロー

具体的には、以下のようになります。

- セッションが存在する場合の処理（高速経路）は変更なし。一度セッションが確立した後は、パケット処理がこれまでと同じように行われる。変更は、初期セッションの確立に使用される低速経路のみ。
- 低速経路では、各 NAT の種類に応じたさまざまなユースケースに対応するために、実行する変換の種類に応じて NAT を行うタイミングが分散される。たとえば、宛先 NAT および静的 NAT は、ルート検索およびセキュリティポリシー処理の前に処理する。それにより、前述のように偽ルートをルーティングテーブルに作成する必要がなくなる。

NAT 処理は、NAT ルールセットの評価を中心にして行われます。NAT ルールセットは、どのようなトラフィックをどのような方式で処理するかを決定するルールの集まりです。ある意味では、NAT ルールセットはセキュリティコンテキスト（ソース / 宛先ゾーンの組み合わせ）とそれほど違いはありません。ルールセットは、処理するトラフィックの全体的な方向性を決定します。たとえば、ルールセットは、特定のインタフェースからのトラフィック、または、指定したゾーンへのトラフィックを選択できます。一方、ルールセット内の各ルールは、ルールに一致するトラフィック（トラフィックのソース / 宛先アドレスでマッチング）と、実行するアクションを指定します。ルールセットには複数のルールを含めることができます（Junos OS 9.3 では1つのルールセットに最大 8 個まで。将来のバージョンではこの制限は変わる可能性があります）。

各 NAT ルールセットは、以下のようなマッチング条件のセットを指定します。

- ソース / 宛先インタフェース
- ソース / 宛先ゾーン
- ソース / 宛先ルーティングインスタンス

NAT ルールセットによっては他よりも指定できるマッチング条件が多い場合もありますが、どのタイプの NAT（ソース、宛先、静的）も設定モデルは同じです。宛先 NAT と静的 NAT はどちらもルート検索よりも前に処理されるため、宛先 NAT ルールおよび静的 NAT ルールは、パケットの宛先ゾーン、インタフェース、またはルーティングインスタンスではマッチングできません。

一致するルールセットが見つかったら、ルールセット内の各ルールに一致するかどうかを評価します。NAT ルールは、以下の要素でマッチングできます。

- ソース / 宛先アドレス
- 宛先ポート（Junos OS バージョン 9.3 現在では、宛先 NAT の場合のみ）

最後に、セッションの確立中にパケットがルールセット内のルールに一致したら、そのルールで指定されているアクションに従ってトラフィックが処理されます。各 NAT タイプの [security nat] 以下の設定階層を以下に示します。

ソースNATの設定階層

```
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>;
                }
            }
        }
}
```

宛先NATの設定階層

```
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>;
                }
            }
        }
}
```

静的NATの設定階層

```
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 プールを使用して、アドレスといくつかの変換パラメータを指定します。プールはソース NAT と宛先 NAT の両方に使用できるため、構成可能なオプションは、必要な NAT のタイプによって異なります。

ソース NAT プールは、以下を指定します。

- アドレス。単一のアドレス、プレフィックス、またはアドレス範囲を指定可能。
- ポート。ポート変換を使用する場合は範囲で指定。ポート変換が不要な場合は "no-translation"（詳細については設定例を参照）。
- アドレスシフティングに使用するホストベースアドレス。
- プールが属するルーティングインスタンス。デフォルトはメインルーティングインスタンス。
- オプションで、プールのアドレスが不足した場合のオーバーフロープール。"no-port-translation" が設定されたプールにのみ使用可能。

宛先 NAT プールは、以下のオプションを指定します。

- 宛先アドレスまたはアドレス範囲。
- 宛先ポート。（図 11 に関連する設定例に示されるような）ポート転送に使用。
- プールが属するルーティングインスタンス。デフォルトはメインルーティングインスタンス。

処理中に、パケットが複数のルールセットに一致する可能性があります。このような曖昧な状況を解決するために、Junos OS は、必ずより詳細に一致する方のルールセットを選択します。具体的には、ルーティングインスタンスよりもゾーン、ゾーンよりもインタフェースの一致の方がより詳細であるとみなされます。

ルールセット内のルールは順番に評価され、パケットに最初に一致したルールが処理に使用されます。ルールセットおよびルールの評価は、以下のように表すことができます。

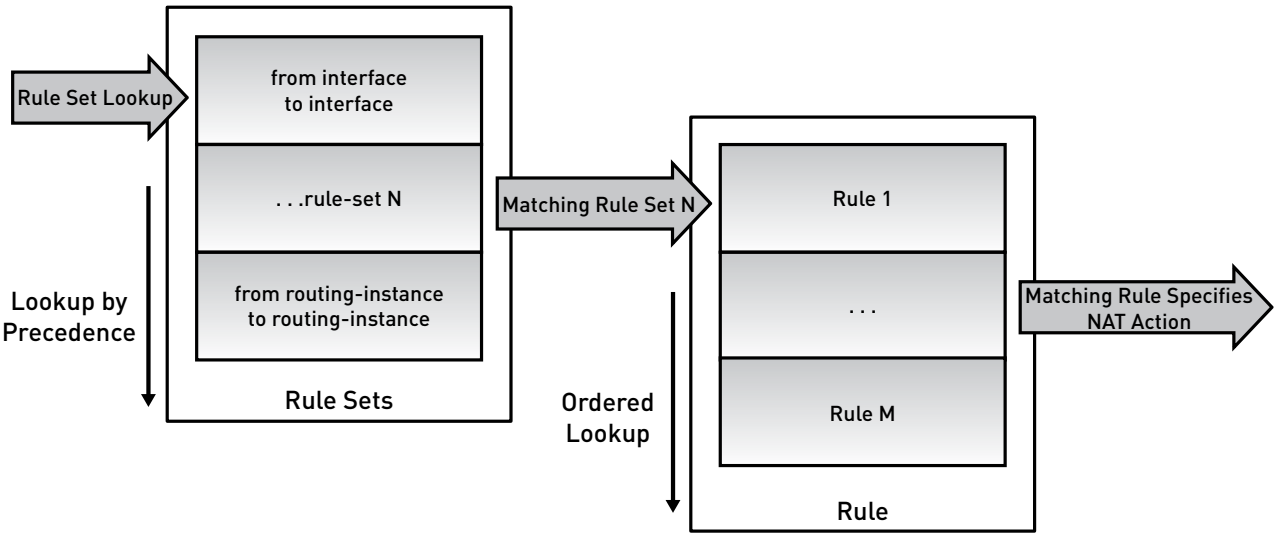


図 3：NAT ルールセット評価のプライオリティ

ソース NAT

ソース NAT は、以下の一般的なアクションを実行するためによく使用されます。

- 単一の IP アドレスを別のアドレスに変換する（私設網を介してインターネットにアクセスする単一デバイスがインターネットにアクセスできるようにするため）。
- 連続したアドレスのブロックを同じサイズの別のブロックに変換する。
- 連続したアドレスのブロックを小さいサイズの別のブロックに変換する。
- ポート変換を用いて、連続したアドレスのブロックを単一の IP または小さいブロックに変換する。
- 連続したアドレスのブロックを出力インタフェースのアドレスに変換する。

同じサイズのアドレスブロックを 1 対 1 および多対多で変換する場合、すべてのアドレスに利用可能なアドレスがプール内に存在するため、ポート変換が必要ありません。アドレスプールのサイズが変換するアドレスの数よりも小さい場合、同時に変換するアドレスの総数をプールのサイズで制限するか（たとえば、デバイスの最大数 253 台がフルに使用されているクラス C をサイズが 10 のプールに変換する場合、一度に接続可能なデバイスの最大数を 10 台にする）、またはポート変換を使用する必要があります。

アクション 5 では出力インタフェースのアドレスを変換に使用するということを除けば、ソース NAT のアクション 4 とアクション 5 は同じです。変換プールには IP が 1 つしか含まれていないため（インタフェースのアドレス）、インタフェース NAT では常にポート変換が使用されます。シナリオ 5 ではアドレスプールは使用しません（このシナリオの構成は図 8 を参照）。

ソース NAT アクションの設定シンタックスを表 1 に示します。

表1：ソースNATのプール構成

シナリオ	構成

単一 IP から単一 IP（ケース 1）	pool <pool name> {
	address <IP address>/32;
	}

多対多、同じサイズのブロック（ケース 2）	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	host-address-base <ip address>
	}

多対多、違うサイズのブロック、ポート変換なし（ケース 3）	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	port no-translation;
	}

多対多、違うサイズのブロック、ポート変換あり（ケース 4）	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	}

宛先 NAT

ソース NAT の場合と同様に、アドレスプールによって変換に使用できるアドレスの数が決まります。これら 2 種類の変換の主な違いは、宛先 NAT では、変換後のアドレスと変換前のアドレスが必ず 1 対 1 で対応するということです。宛先 NAT は、以下の一般的なアクションを実行するためによく使用されます。

1. 単一の IP アドレスを別の IP アドレスに変換する。インターネット上の複数のデバイスが私設網内の 1 台のホストに接続できるようにするためによく使用される。
2. 連続したアドレスのブロックを同じサイズの別のブロックに変換する。サーバーグループにアクセスできるようにするために使用できる。
3. 宛先 IP/ ポートの組み合わせを別の宛先 IP/ ポートに変換する。同じ IP アドレスを使用するがポートが異なる複数のサービスにアクセスできるようにするために使用される場合がある（図 11 を参照）。

宛先 NAT アクションの設定シンタックスを表 2 に示します。

表2：宛先NATのプール構成

シナリオ	構成

単一 IP から単一 IP（ケース 1）	pool <pool name> {
	address <IP address>/32;
	}

多対多、同じサイズのブロック（ケース 2）	pool <pool name> {
	address <address-low> to <address-high> | <network>/<netmask>;
	}

特定の IP/ ポートの組み合わせから別の IP/ ポート	pool <pool name> {
	address <IP address>/32;
	port <port number>;
	}

静的 NAT

前のケースでは、ソース NAT と宛先 NAT のどちらも、ネットワークの一方からのみ接続を開始できます。ソース NAT は外方向への接続のみに作用します（私設網からインターネットへの接続など）、宛先 NAT は内方向への接続のみに作用します（インターネットから私設網上のサーバーへの接続など）。

静的 NAT は、ネットワークのどちらの側からでも接続を開始できますが、1 対 1 の変換（もしくは、ブロックのサイズが同じである多対多の変換）が可能な場合のみに限定されます。つまり、どちらの側からでも接続を開始できるようにするには、各プライベートアドレスにパブリックアドレスを割り当てる必要があります。

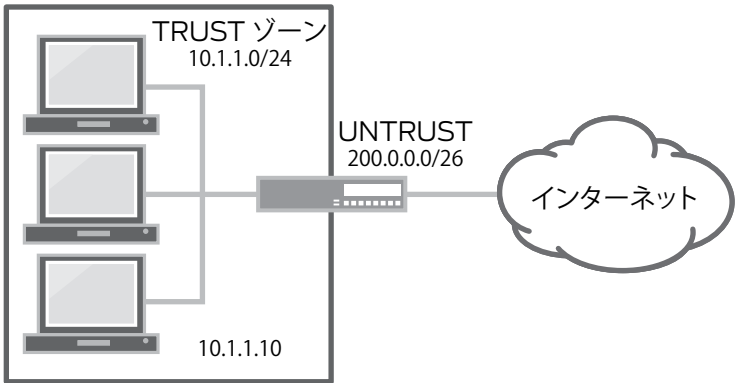
導入の組み合わせが非常に単純であるため、アドレスプールは必要ありません。マッチングルールごとに単一のプレフィックスを変換に使用する必要があり、また、マッチングプレフィックスと変換プレフィックスが同じサイズでなければなりません。

設定例

以下のセクションでは、これまでに説明したさまざまな導入シナリオの設定例をいくつか紹介します。ほとんどの例が同じ構造に従っています。各例には、ネットワーク設計例、変換後と変換前のアドレス、および必要な設定が含まれています。

ソース NAT の設定例

ソースNAT — 単一アドレス変換

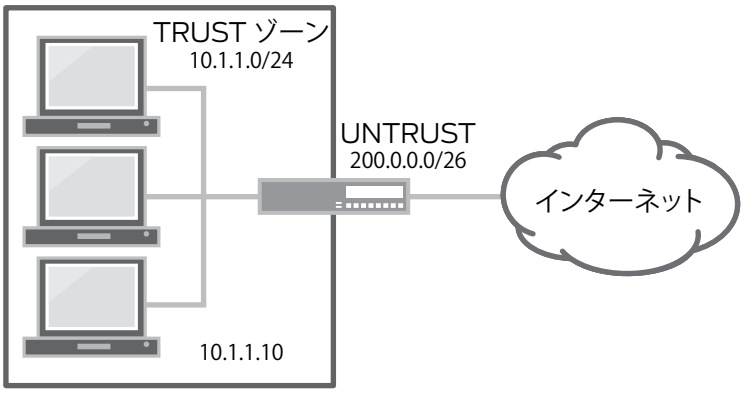


元のソース	変換後のソース
10.1.1.10	200.0.0.10

図 4：単一アドレス変換

```
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;
            }
        }
    }
}
```

ソースNAT — アドレスブロックのサイズが同一の多対多変換



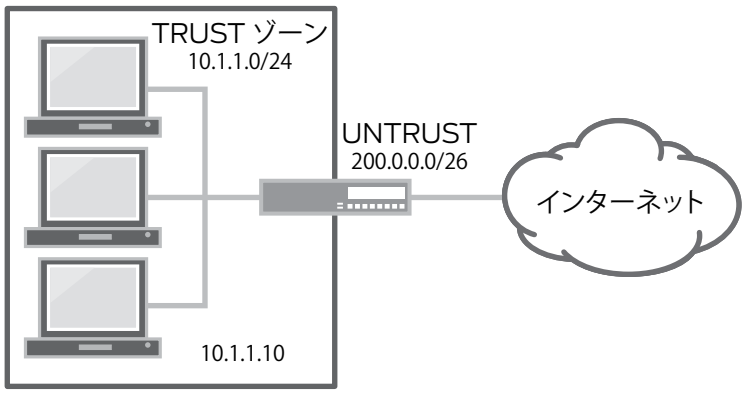
元のソース	変換後のソース
10.1.1.10 to 10.1.1.20	200.0.0.30 to 200.0.0.40

図 5：アドレスシフティング

```
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;
            }
        }
    }
}
```

この例は考察に値します。host-address-base オプションには、アドレス範囲の開始点となる元のソースアドレスを指定します。ルールセットのマッチング条件ではアドレス範囲の指定が許可されないため（プレフィックスのみ）、このオプションは必須となります。ルールの source-address に一致するが、[host-address-base, host-address-base+pool size] の範囲内でないソースアドレスを持つパケットは、変換されません。

ソースNAT — アドレスブロックのサイズが異なる多対多変換

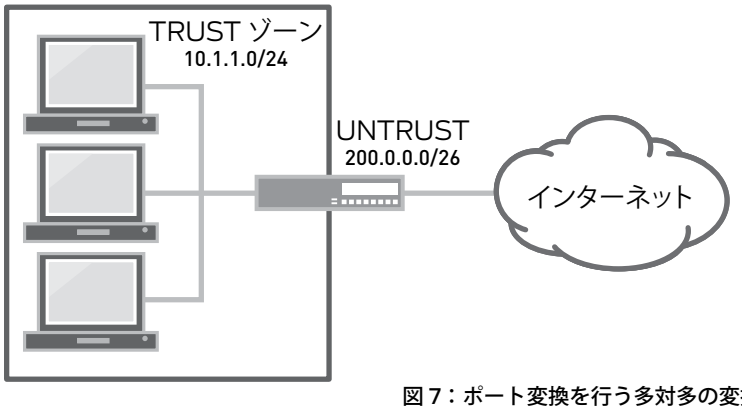


元のソース	変換後のソース
10.1.1.0/24	200.0.0.30 to 200.0.0.40

図 6：ポート変換を行わない多対多の変換

```
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;
            }
        }
    }
}
```

ソースNAT — ポート変換を行う、アドレスブロックのサイズが異なる多対多変換

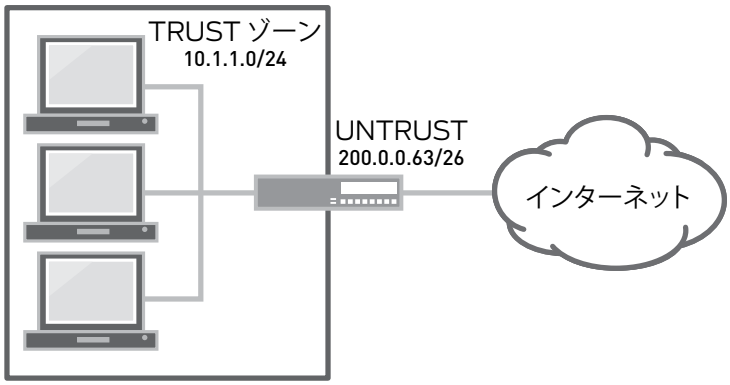


元のソース	変換後のソース
10.1.1.0/24	200.0.0.1 to 200.0.0.5

図 7：ポート変換を行う多対多の変換

```
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;
            }
        }
    }
}
```

ソースNAT — インタフェースNAT



元の ソース	変換後の ソース
10.1.1.0/24	Interface IP (200.0.0.63)

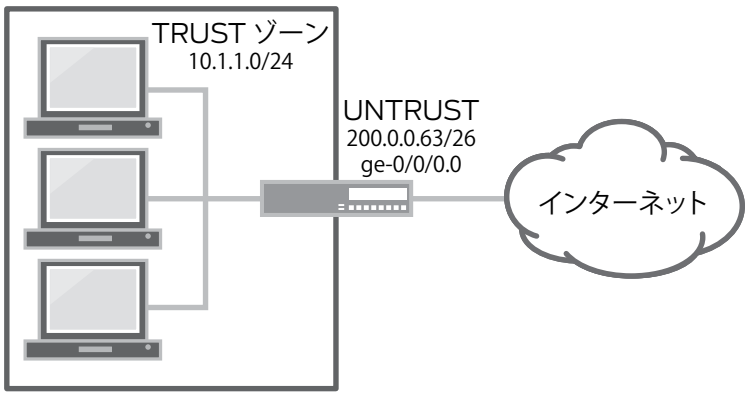
図 8：インタフェース NAT

```
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;
            }
        }
    }
}
```

"source-nat interface" というキーワードには、2 つの機能があります。1 つは、パケットを出力インタフェースのアドレスに変換できるようになります。もう 1 つは、出力インタフェースのアドレスがわからない場合でも NAT が可能になります。たとえば、DHCP（Dynamic Host Configuration Protocol）または PPP（Point-to-Point Protocol）によってアドレスが動的に割り当てられる場合などです。

宛先 NAT の設定例

宛先NAT — 単一アドレス変換



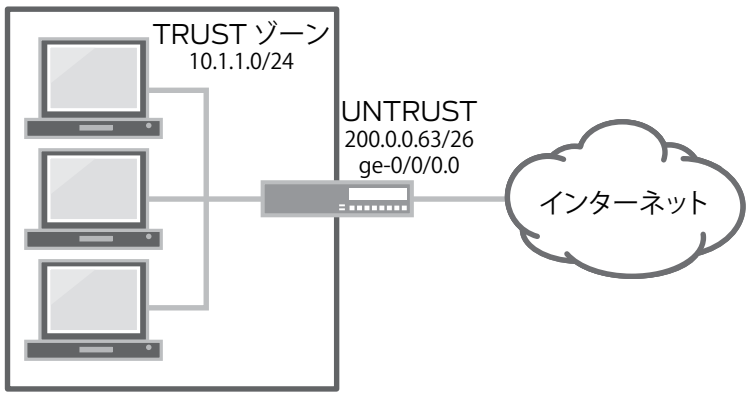
元の 宛先 IP	変換後の 宛先 IP
200.0.0.11	10.1.1.11

図 9：単一アドレスの宛先変換

```
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;
        }
    }
}
```

この例では、アドレス 200.0.0.11 は、Untrust インタフェース ge-0/0/0.0 サブネットに属します。このインタフェースが、アドレス 200.0.0.11（10.1.1.11 に変換されるパブリックアドレス）への ARP（Address Resolution Protocol）要求に応答するには、例のように、インタフェース上に proxy-arp を構成する必要があります。

宛先NAT — 多対多変換



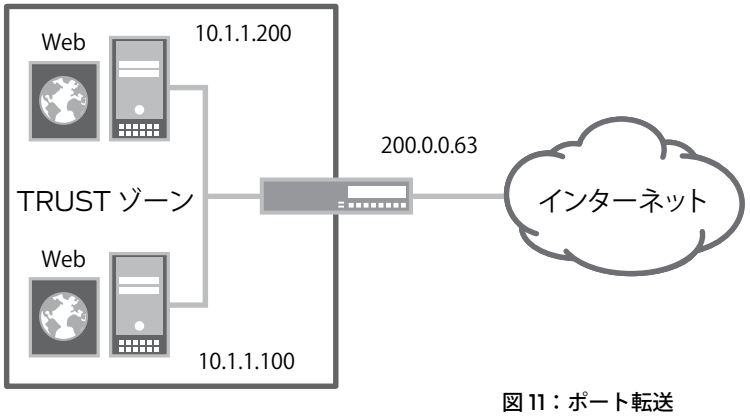
元の宛先 IP	変換後の宛先 IP
200.0.0.0/26	10.1.1.0/26

図 10：多対多の宛先変換

```
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;
        }
    }
}
```

前の例と同様に、NAT プールは ge-0/0/0.0 インタフェースのアドレス範囲に属します。そのため、インタフェース自体の IP アドレスを除くプール内のすべてのアドレスに対してプロキシ ARP が有効です。

宛先NAT — IP/ポート変換



元の宛先 IP	変換後の宛先 IP
200.0.0.63 ポート80	10.1.1.100 ポート80
200.0.0.63 ポート8080	10.1.1.200 ポート80

図 11：ポート転送

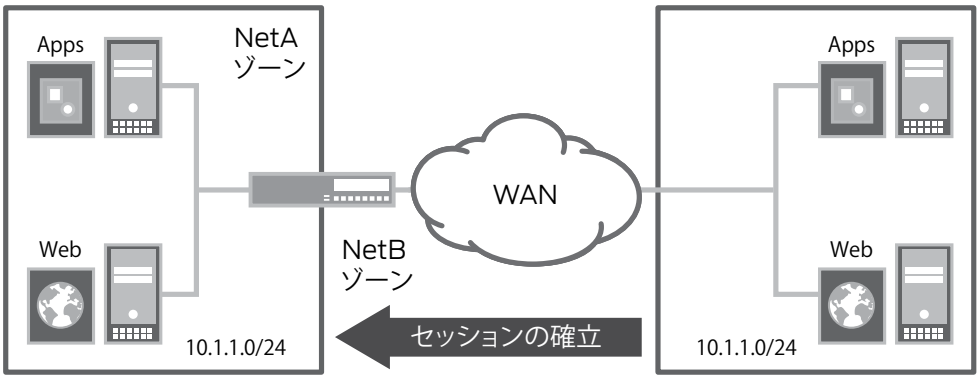
```
destination {
    pool server-1 {
        address 10.1.1.100/32 port 80;
    }
    pool server-2 {
        address 10.1.1.200/32 port 80;
    }
    rule-set nat-example {
        from interface ge-0/0/0.0;
        rule port-forwarding {
            match {
                destination-address 200.0.0.63/32;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

静的 NAT の設定例

この例では、図 10 で紹介した設定を再び行いますが、代わりに静的 NAT を使用します。この例の唯一違うところは、新しい設定により、ゲートウェイのどちら側からでも接続できるようになることです。宛先 NAT の多対多変換では、インターネットから 10.1.1.0/26 ネットワークへの接続のみ確立可能でした。

```
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;
        }
    }
}
```

デュアル NAT



元の ソース IP	変換後の ソース IP
10.1.1.0/24	10.2.0.0/24
元の 宛先 IP	変換後の 宛先 IP
10.2.0.0/24	10.1.1.0/24

図 12：デュアル NAT

この例では、ソースアドレスと宛先アドレスの両方が同時に変換されます。一般に、このような例は、アドレス空間衝突が発生しているネットワークで見られます。NetB 上のホストが NetA 上のホストに接続したい場合、10.2.0.0/24 ネットワークにパケットを送信することによって接続を確立します。たとえば、NetB 上のホスト 10.1.1.1 が NetA 上のホスト 10.1.1.5 とのセッションを確立したい場合、ホスト 10.1.1.1 はアドレス 10.2.0.5 にパケットを送信します。10.1.1.0/24 ネットワークから送られてきた 10.2.0.0/24 ネットワークの受信パケットは、10.2.0.0/24 ネットワークから送られる 10.1.1.0/24 ネットワーク行きのパケットとして変換されます。それにより、両端のホストが 10.2.0.0/24 ネットワークを中継として使って通信できるようになります。

```
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;
            }
        }
    }
}
```

NAT とセキュリティポリシーの相互作用

NAT 処理がセキュリティポリシーから分離されたため、変換後のアドレスと変換前のアドレスが両方ともポリシーに一致する場合があります。

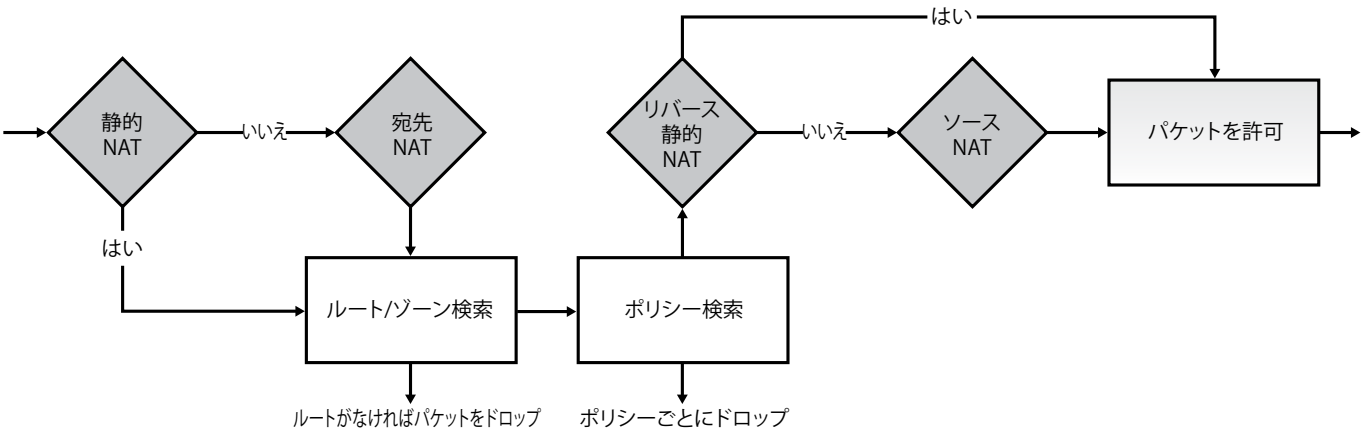


図 13：ポリシーと NAT の処理

図13は、NAT処理を簡素化して表しています（図2の新しいパケットフローにも示されています）。この図から、以下のことがわかります。

- ポリシーエンジンは、宛先 NAT または静的 NAT の変換後のアドレスをマッチングする。つまり、パケットはポリシーエンジンに入る前に変換される。
- ポリシーエンジンは、ソース NAT またはリバース静的 NAT 接続の変換前のアドレスをマッチングする。

設計上では、ファイアウォール管理者が気にする必要があるのは、通信に関わる実際のエンドポイントについてだけです。インターネット上のサーバーとの接続（ソース NAT を介して）を確立している私設網上のクライアントは、クライアントのプライベートアドレスとサーバーの（パブリック）アドレスに一致するポリシーが必要になります。インターネットホストから私設網内のサーバーへの接続（宛先 NAT を使用したパブリック IP による）でも、クライアントアドレスとサーバーのプライベート（変換後の）アドレスに一致するポリシーが必要になります。

しかし、はっきりしない場合もあります。以下のようなネットワークがあるとします。

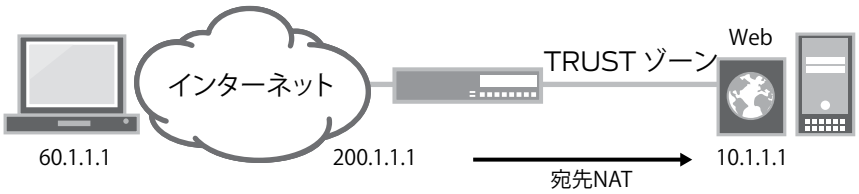


図 14：非 NAT トラフィックのドロップ

このネットワークでは、Trust ゾーン内の Web サーバー（アドレス 10.1.1.1）へのインターネット接続を許可するために、ゲートウェイが宛先 NAT を実行します。すでに述べたように、クライアント（この例では 60.1.1.1）からサーバー 10.1.1.1 の変換後のアドレスへのトラフィックをポリシーエンジンが監視します。問題は、ポリシーエンジンにとって、プライベートアドレス 10.1.1.1 宛てに送信されたトラフィックと、10.1.1.1 に変換されたアドレス 200.1.1.1 宛てのトラフィックとの区別がつかないことです。つまり、サーバーのパブリックアドレス宛てに送信されたトラフィック（有効）と、サーバーのプライベートアドレス宛てに送信されたトラフィック（無効）を、ポリシーエンジンは同一と判断してしまうのです。この問題を解決するために、宛先 NAT または静的 NAT が使用された場合は、トラフィックが変換されていることをポリシーエンジンに知らせます。それにより、以下の例のように、変換後または変換前のトラフィックをポリシーが明確にドロップできるようになります。

```
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;
                }
            }
        }
    }
}
```

監視

Junos OS には、NAT の検証、監視、およびトラブルシューティングに使用できるコマンドがあります。

まず最初に確認すると良いのが、セッションテーブルです。セッションテーブルには、デバイスで現在追跡されているすべてのセッションがリストされます。セッションテーブルには、パケットのソース / 宛先アドレス、ソース / 宛先ゾーン、ソース / 宛先ポート、プロトコル、およびトラフィックが変換されている場合は変換情報が表示されます。

セッションテーブルは、"show security flow session" コマンドを使って表示します。このコマンドでは、テーブルデータに基づいてセッションをフィルタするオプションを使用できます。

```
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
```

上の出力には、nat-example-security-policy という名前のポリシーに一致するトラフィックが示されています。"In" 命令は、ゲートウェイが受信したトラフィックをそのまま示しています（つまり、変換される前）。"Out" 命令は、出力されることが予想されるトラフィックを示しています（変換された後）。X から Y への接続は、X' to Y' に変換され、"In" 命令では X->Y、"Out" 命令では Y'->X' と示されます。したがって、管理者は、すべてのセッションだけでなくすべての変換についても把握できます。

この例では、207.17.137.228 と 80 のアドレス / ポートの組み合わせ（おそらく、HTTP トラフィックを示す）を使用して、クライアント 10.1.0.13 からサーバーへの接続が確立された TCP セッションを確認できます。ソースアドレスは、10.1.0.13 ポート 52939 から 172.19.101.42 ポート 2132 に変換されました（ポート変換有りのソース NAT）。

"show security nat source|destination|static rule all" も便利なコマンドです。このコマンドは、構成されているすべての NAT ルールと変換が行われた回数（各ルールによって変換されたセッションの数）を示します。

```
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
```

"show security nat interface-nat-ports" コマンドは、インタフェースベースの NAT に使用されているすべての割り当てポートを表示します。最後に、"show security nat source|destination pool" コマンドを発行すると、構成されているプール、プールのサイズ、および変換が行われた回数が表示されます。

```
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
```

まとめ

ジュニパーネットワークスの SRX シリーズ サービス・ゲートウェイでは、Junos OS リリース 9.2 以降、NAT を再構築し、セキュリティポリシーから NAT を分離しました。Junos OS リリース 9.5 より、J シリーズ サービスルーターにも同様の改良が移植されます。この新しいアーキテクチャは、厳しいネットワーク要件に対応できる柔軟性を持ちながら、使いやすさも向上させます。

ジュニパーネットワークスについて

ジュニパーネットワークスは、ネットワークイノベーション企業です。デバイスからデータセンター、消費者からクラウド事業者にいたるまで、ジュニパーネットワークスは、ネットワーキング体験とビジネスを変革するソフトウェア、シリコン、システムを提供しています。ジュニパーネットワークスに関する詳細な情報は、以下をご覧ください。

http://www.juniper.net/jp/ 、Twitter 、Facebook

日本
ジュニパーネットワークス株式会社

東京本社
〒 163-1445
東京都新宿区西新宿 3-20-2
東京オペラシティタワー 45F

電話　03-5333-7400
FAX　03-5333-7401

西日本事務所
〒 541-0041
大阪府大阪市中央区北浜 1-1-27
グランクリュ大阪北浜

URL http://www.juniper.net/jp/

米国本社
Juniper Networks, Inc.

1194 North Mathilda Ave
Sunnyvale, CA 94089
USA

電話　888-JUNIPER
(888-586-4737)
または 408-745-2000
FAX　408-745-2100

URL　http://www.juniper.net

アジアパシフィック、ヨーロッパ、中東、アフリカ
Juniper Networks International B.V.

Boeing Avenue 240
1119 PZ Schiphol-Rijk
Amsterdam, The Netherlands

電話　31-0-207-125-700
FAX　31-0-207-125-701

3500151-002 JP Apr 2014