

アプリケーションノート

JUNOSルーターセキュリティ

実例に学ぶ強いインフラづくり

ペジヤン・ペイマニ
システムエンジニア

マット・コロソ
マーケティングエンジニア



Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
408 745 2000 or 888 JUNIPER
www.juniper.net

ジュニパーネットワークス株式会社
〒163-1035 東京都新宿区西新宿3-7-1
新宿パークタワー N棟 35階
電話 03-5321-2600
FAX 03-5321-2700
URL <http://www.juniper.co.jp>

Part Number : 350013-003JP 06/03

目次

概要	3
背景	3
ルーターセキュリティ	3
JUNOSのデフォルト設定	4
ルーターアクセスセキュリティ	5
帯域外管理	5
帯域内管理	5
ルーターとの通信	5
ssh (Secure Shell)	5
scp (Secure Copy Protocol)	6
認証の一元化	6
ルーティングプロトコルセキュリティ	8
ルーティングエンジンが受信するトラフィックのフィルタリング	9
ルーティングエンジンのトラフィックにファイアウォールフィルタを適用	9
ファイアウォールフィルタの設計	9
ICMPフラッド攻撃とSYNフラッド攻撃への対処	10
悪意あるフラグメンテーションへの対処	11
IPオプションを持つパケットへの対処	12
ルーターを攻撃のリフレクタに利用されないための対策	13
VPNを巡る課題	14
各VPNへのファイアウォールフィルタの適用	15
ルーティングエンジンへのトラフィックのレートリミット	15
セキュリティのための監査	16
認証・コマンドのイベントログ作成	16
ルーティングプロトコルのイベントとエラー	16
拒否されたトラフィックのログ作成	17
NTP (Network Time Protocol)	17
まとめ	17
付録A	18
付録B	18

概要

IPとインターネットの特徴である匿名性や柔軟性も、悪意のある攻撃者にとっては、遠く離れた場所からルーター性能を低下させたり、多数のエンドユーザーの通信を妨害したりするのに好都合な道具になってしまいます。サービスプロバイダのルーターをねらった攻撃が増えている今、何らかの防御策が欠かせません。本アプリケーションペーパーでは、ルーターの設定・運用に伴うセキュリティ上の共通課題を取り上げたいので、攻撃の防止・対処に役立つジュニパーネットワークス製ルーターの各種機能について解説します。また、ジュニパーネットワークス製ルーターの安全な導入・運用の参考となる具体例を紹介いたします。

なお、今回ご紹介する設定・運用テクニックは、ルーターのセキュリティを実現する唯一の方法ではありませんし、ジュニパーネットワークス環境に必須というわけでもありません。むしろ、お客様やパートナー各社との経験を基に共通ノウハウとしてぜひお勧めしたい成功事例と言えます。

背景

ルーターをはじめ、各種ネットワーク構成要素に対する攻撃は、公表されているものだけを見て増加の一途をたどっています。こうした動きに呼応するように、ネットワークサービスプロバイダ向けのきわめて高度なセキュリティ環境を構築するため、あらゆる公共インフラのセキュリティに注目が集まっています。ネットワークセキュリティ問題にはさまざまな側面があり、ルーターの設定やルーター型ネットワークの設計に関わってきます。そこで、サービスプロバイダとしては、妨害や破壊を目的とした攻撃からネットワークを守るため、ネットワークやルーターの設計特性を見極め、積極的に活用することで、ネットワークのセキュリティ強化につなげる必要があります。

注 本資料で紹介する方法は、ルーターインフラのセキュリティ強化を支援するものであって、総合的なネットワークセキュリティ戦略のごく一部にすぎません。また、ネットワークサービスには、ネットワーク管理、ドメインネーム解決、ユーザーとのメール交換などの各種機能を担うホストシステムも必要です。本資料に登場する手法の中には、ルーターだけでなくホストの保護にも応用できるものもありますが、具体例としては、ルーターセキュリティに特化した手法となっています。

ルーターセキュリティ

ルーターセキュリティは、次の3つの要素に分けることができます。

- ルーターの物理的なセキュリティ
- オペレーティングシステムのセキュリティ
- 設定の強化

本資料には、物理的なセキュリティに関する推奨事項は特に盛り込まれていませんが、ネットワーク機器に物理的に近づかせないことがセキュリティの第一歩であることは言うまでもありません。遠隔地からであれば簡単に防御できる攻撃であっても、管理ポートやコンソールに直接接続されてしまえば、攻撃を防ぐことは非常に困難です。

ルーターのオペレーティングシステムに固有のセキュリティは、ルーター自体のセキュリティでも重要な役割を果たします。念入りにルーターを安全な設定にしても、オペレーティングシステムそのものの安全性が確保されていなければ、結局、ルーターが危険にさらされます。たとえば、ルーターを攻撃するハッカーが共通して使うテクニックの1つに、ルーターのオペレーティングシステムコードの穴を見つけだし、バッファあふれを引き起こす方法があります。このような悪用を防止するため、オペレーティングシステムには、高度な安定性と堅牢性が欠かせません。ジュニパーネットワークスでは、ソフトウェアのリリース時に安定性確認やバグの解消に細心の注意を払っていますが、これに加えて、きわめて希なソフトウェア障害が発生した場合でもシステムが攻撃されないような方法とガイドラインを用意しています。安全なオペレーティングシステムの条件として、システムを攻撃から保護する機能が搭載されていることが挙げられます。専門知識のあるユーザーであれば、さまざまなエレメントの採用によって発生する脆弱性を最小限に抑えることができます。

本資料では、先ほどルーターセキュリティの3番目の要素として挙げた「設定の強化」を中心に解説します。設定の強化とは、ルーターのオペレーティングシステムで利用可能なツールを駆使して、しっかりしたセキュリティポリシーを適用することにはかなりません。堅牢なセキュリティツールがあれば、基本的にどのような設定でも、安全に運用できます。しかし、このようなツールがあっても、完璧なセキュリティを誇るオペレーティングシステムの設定を間違えれば、ルーターが脆弱になることもあります。

さらに、本資料の後半では、設定強化に関する4つの要素について解説します。

- ルーターへのアクセスのセキュリティ

- ルーティングプロトコルのセキュリティ
- ルーティングエンジンの保護
- セキュリティ監査

本資料で取り上げるセキュリティの推奨事項や検討事項は、IPv4環境を前提としています。実際には、IPv6環境でも、同じような問題点や対応策が多数ありますが、今回は具体的な形で取り上げません。

注 本資料の後半の内容に関しては、JUNOSソフトウェアの設定機能やシンタックスに慣れ親しんでいる方々を対象としています。したがって、掲載されているコマンドや設定オプションの使用に当たっては、その結果について十分理解されているものとし、使用方法を誤った場合、トラフィックフローの抑制やルーターへのログイン不能といった事態になり、ネットワーク停止に至ることもありますので、ご注意ください。詳しい設定や操作コマンドについては、www.juniper.net/techpubs/にある技術資料をご覧ください。

JUNOSのデフォルト設定

JUNOSソフトウェアをインストールし、ルートアカウントのパスワードを設定した瞬間から、攻撃しにくい環境が生まれます。まず、以下に挙げたようなルーター関連の一般的な問題点は、いずれもJUNOSソフトウェアがデフォルト設定で対処できるものばかりです。

- JUNOSソフトウェアは、directed broadcastパケットを転送しません。directed broadcastサービスは、200.0.0.255などのブロードキャストアドレスに偽装ソースアドレスからping要求を送り、インターネット上の他のユーザーを攻撃する際に悪用されることがあります。こうした「ブロードキャストのping」を200.0.0.0/24のネットワークで許可すると、1回のping要求で最大254の応答を引き出せます。応答はすべてpingを要求した偽装アドレスに送られるため、DoS攻撃の餌食になってしまうのです。大規模ネットワークや複数のネットワークともなれば、影響はさらに大きくなります。
- デフォルトでは、遠隔管理を目的としたルーターへのアクセスは、使用不可に設定されています。有効にするためには明示的に使用可に設定しなければなりません。デフォルト設定では、コンソールによるルーターへのアクセスだけが許可されています。この設定は、telnetやftp、さらにはsshなど、あらゆる管理アクセスプロトコルに適用されます。
- JUNOSソフトウェアは、設定データの編集に利用するSNMP設定機能をサポートしていません。ネットワークの監視・トラブルシューティングを目的としたSNMP設定機能はサポートしていますが、これによって既知のセキュリティ問題に発展することはありませんし、必要があれば、使用不可に設定できます。
- JUNOSソフトウェアは、デフォルトでは、ルーティングエンジンに送られるARPパケットにレートリミット機能を適用します。この機能では、設定ミスや不正行為による「ARPストーム」攻撃を阻止できます。さらにARPメッセージにもレートリミットを適用することで、ARPポリサーをイーサネットインタフェースに設定することも可能です。
- いわゆるmartianアドレス（偽アドレス）は、ホストやネットワークの予約アドレスになっているため、ルーティング情報は完全に無視されます。デフォルトのmartianアドレスリストに具体的なアドレスを追加すると、お使いのルーターの設定に次の情報が設定されます。

```
routing-options {
    martians {
        destination-prefix match-type;
    }
}
```

JUNOSソフトウェアは、デフォルト設定で、次のプリフィクスがmartianアドレスに登録されています。

```
0.0.0.0/8
127.0.0.0/8
128.0.0.0/16
191.255.0.0/16
192.0.0.0/24
223.255.255.0/24
240.0.0.0/4
```

ルーターアクセスセキュリティ

ルーター用インフラの管理・保守には、安全なリモートアクセスが欠かせません。前述のように、JUNOSソフトウェアは、デフォルトでリモートアクセスがすべて使用不可にされているため、最初からリモートアクセスのセキュリティが確保されています。このため、正規ユーザーが意識的に機能を設定しないかぎり、リモートアクセスは実行できません。ルーターとのリモート通信を確立する方法としては、帯域外通信と帯域内通信の2種類があります。

帯域外管理

帯域外方式の場合、ルーター管理専用のインタフェースを使ってルーターに接続できます。ジュニパーネットワークス製ルーターは、管理専用イーサネットインタフェース (fxp0) やEIA-232のコンソールポート・補助ポートによる帯域外管理に対応しています。管理用のイーサネットインタフェースは、ルーティングエンジンに直接接続できます。このインタフェースを経由する中継トラフィックは許可されていないため、ユーザートラフィックと管理用トラフィックを完全に分離でき、中継ネットワークで輻輳や障害が発生した場合でも、ルーターの管理に影響が及ぶことはありません。管理ネットワークを別途用意することで、DoS攻撃などによるサービス停止時でもルーターインフラを簡単に管理できます。

大手ISPの多くは、現用系ネットワークに破滅的な障害が発生した場合でも遠隔地から管理・復旧作業を実施できるように、帯域外の管理ネットワークを保有しています。この方式はコストがかかり、多大なリソースが必要になりますが、管理制御用のトラフィックと中継用トラフィックを明確に分離できる利点があります。しかし、帯域外ネットワーク自体に障害が発生することもあるので、注意が必要です。

帯域内管理

帯域内管理方式では、ユーザートラフィック用のインタフェースと同じインタフェースを利用してルーターに接続します。この方式は仕組みが単純で専用の管理リソースも不要ですが、デメリットがいくつかあります。

- 管理用と中継用のトラフィックフローが混在しているため、通常のトラフィックに攻撃用のトラフィックが混入すると、ルーターとの通信機能が影響を受ける恐れがあります。
- ルーター間のリンクの信頼性が低下する恐れがあり、盗聴やリプレイ攻撃の標的にされやすくなります。

ほとんどのネットワークでは、帯域内でルーティングプロトコルのトラフィックが流れる形態のため、帯域内管理トラフィックの場合と同様のセキュリティ対策が必要です。そこで、本資料の後半では、ルーターとの管理用通信の方式として、帯域内管理方式を想定しています。

ルーターとの通信

リモートコンソールからルーターに接続して通信する場合、管理用のアクセス方式は数種類あります。最も一般的なのは、telnetやssh (secure shell) のアプリケーションを使う方法です。ただし、当社では、次のような理由から、コンソールによるアクセスは、telnetではなくsshを推奨しています。

ssh (Secure Shell)

ssh (secure shell) は、セキュリティ保護されていないネットワークで安全な暗号化通信を実現できるため、帯域内ルーター管理に威力を発揮します。しかし、JUNOSソフトウェアでは、sshによるルーターへのアクセスは、他のネットワークアクセス方式と同様に、デフォルト設定で使用不可になっています。JUNOSでsshを使用可能に設定すれば、sshによるアクセスが可能になり、1分間で確立可能なssh同時セッション数や最大sshセッション数の制御に使うオプションのパラメータを設定できます。また、rate-limitパラメータは、sshポートに対するSYNフラッド攻撃の防止に効果的です。

```
system {
  services {
    ssh connection-limit 10 rate-limit 4;
  }
}
```

sshを有効にすると、全ユーザーがルーターのアクセスにsshを利用できます。ただし、sshによるルートアカウントへのアクセスは、以下の設定で制限できます。

```
system {
  services {
    ssh {
```

```
        root-login deny;  
        protocol-version v2;  
    }  
}
```

denyオプションは、sshによる一切のルートアクセスを無効にします。もう1つのdeny-passwordオプションは、ルートユーザのパスワードが設定されていない場合にだけ、ルートアクセスを許可します。deny-passwordオプションは、配備・インストール時に初めてルーターにアクセスする場合に便利です。

デフォルト設定では、ルーターはsshのバージョン1とバージョン2の両方の接続要求を受け入れます。どちらかのバージョンに限定することもできます。上記の設定では、sshのバージョン1を使ったアクセスが許可されません。

通常、sshは、バージョン1よりバージョン2のほうが安全性が高いと考えられています。これは、バージョン2のほうが設計上の配慮が行き届いていて、マニュアルも充実しており、認知度も高いことが理由です。そこで、当社としては、なるべくバージョン2の利用を推奨しています。

scp (Secure Copy Protocol)

scp (secure copy protocol) は、sshによる暗号化・認証基盤を利用して、ホスト間で安全にファイルコピーを実行します。JUNOSソフトウェアでは、運用モードのfile copyコマンドでscpを利用できます。たとえば、次の運用モードのコマンドをご覧ください。

```
file copy router.config user@remoterouter.example.com/destdir/
```

このコマンドは、ローカルのカレントディレクトリにあるrouter.configというファイルをremoterouter.example.comにあるdestdirディレクトリにコピーします。特に中継パスが信頼性の低いネットワークを通過する設定の場合、ルーター間のファイルコピーには、ftpの代わりになるべくscpを使ってください。

認証の一元化

多数のルーターを複数の担当で管理していると、ユーザーアカウント管理面で深刻な問題が発生することがあります。この問題には、認証の一元化サービスを利用してアカウント管理を簡素化することで対処できます。この方式では、アカウントの作成・削除は、中央の特定のサーバーだけで実行されます。たとえば、全ルーターへの従業員のアクセスを有効または無効に設定するときも、1回の変更だけで済み、ルーターに対する設定変更の確定さえも不要になります。

また、集中型の認証システムでは、SecureIDのようなワンタイムパスワードシステムが簡単に扱えるため、パスワードスニフィングやパスワードリプレイ攻撃（キャプチャしたパスワードを悪用してルーター管理者になりすます攻撃）に対する有効な防御策を実施できます。ワンタイムパスワード方式の場合、正規ユーザーがシステムにアクセスするたびに、異なるパスワード文字列が使われます。このため、仮に攻撃者がパスワードをキャプチャできたとしても、その後のログインでシステムが受け付けないのです。

JUNOSソフトウェアは、複数のルーターでのユーザーの集中認証に、RADIUS (Remote Authentication Dial In User Service) とTACACS+ (Terminal Access Controller Access Control System Plus) の2種類のプロトコルをサポートしています。

当社では、認証サービスプロトコルにRADIUSを推奨しています。RADIUSは、マルチベンダ環境に対応したIETF策定の標準規格であり、機能的にもTACACS+などベンダ独自仕様の規格に比べて幅広く普及しています。さらに、セキュリティ強化のため、ワンタイムパスワードシステムの併用をお勧めします。ちなみに、ワンタイムパスワードシステムのベンダは例外なくRADIUSをサポートしています。

JUNOSの集中認証モデルは、基本的にRADIUSやTACACS+と同じです。ルーターにアカウントテンプレートをいくつでも作成できるため、ユーザーがルーターにアクセスする際に、アカウントテンプレートを使用するように設定できます。そこで、2種類のサーバーで3段階のアクセスレベルをサポートする設定例を以下に挙げておきます。

```
system {  
    authentication-order [ radius ];  
    radius-server {  
        10.1.2.1 {
```

```

        secret "XXXXXXXXXXXXXXXXXXXX"; # SECRET-DATA
        timeout 5;
    }
    10.1.2.2{
        secret "XXXXXXXXXXXXXXXXXXXX"; # SECRET-DATA
        timeout 5;
    }
}
}

```

上記のauthentication-orderコマンドで、RADIUS認証が有効になります。RADIUSサーバーにアクセスできない場合に限り、フォールバックモード（ルーターにローカルアカウントをセットアップして目的を達成する仕組み）に入ります。

2つのサーバーセクションがあるため、RADIUSプロトコルが有効になり、クライアント（ルーター）とサーバーの間で共有する秘密情報を定義し、双方ともに通信相手のピアが信頼できることを確認します。また、サーバーごとにタイムアウトを設定し、指定時間内にレスポンスがない場合には、ルーターが接続対象を別のサーバ（第1セクション）に変更するか、別の認証方式（第2セクション）に切り替えることができます。

次に、複数のユーザークラスを作成し、それぞれに特権を設定できます。ここでもタイムアウトを使って、非アクティブのまま一定時間が経過後にクラスメンバとの接続を切断する例を紹介します。もちろん、ユーザーの特権レベル（とクラスメンバ）は、本来、組織内での任務に依存するものであって、ここで取り上げる権限はあくまでも一例に過ぎません。

```

login {
/* このユーザークラスは、統計情報と設定しか閲覧できません。設定を修正する権限はありません。*/
    class observation {
        idle-timeout 5;
        permissions [ configure firewall interface network routing
            snmp system trace view];
    }
/* このユーザークラスは、設定の閲覧と修正が可能です。*/
    class operation {
        idle-timeout 5;
        permissions [admin clear configure interface
            interface-control network reset routing routing-control snmp
            snmp-control trace-control firewall-control rollback];
    }
/* このユーザークラスは、アクセス権と制御権に制約がありません */
    class engineering {
        idle-timeout 5;
        permissions all;
    }
}
}

```

クラスとそれぞれの権限の定義が終われば、ローカルのスーパーユーザーとして、RADIUS認証で使用するクラスに対応するユーザーを定義できるようになります。

RSA/DSA鍵を使った公開鍵認証は、単純なパスワードの交換と比べて、はるかに安全性に優れています。単純なパスワードは、辞書攻撃などで推測・予測されやすいといった問題があり危険です。RSA/DSA鍵であれば、公開暗号鍵による認証を実施するため、管理用のアクセスにtelnetを使わずに済む点も、単純なパスワード方式にはないメリットと言えます。

```

login {
/* これはローカルのスーパーユーザーのアカウントです。RADIUSの障害時やアクセス不能時には、ルーターのローカルアカウント使用に戻ります。*/
    user admin {
        uid 1000;
        class engineering;
        authentication {
            ssh-dsa "XXXXXXXXXXXXXXXXXXXX"; # Secure shell (ssh) RSA public
key string

```

```

    }
  }
/* ここでは、3種類のユーザーまたはユーザーグループのRADIUSテンプレートを定義します。*/
  user observation {
    uid 1001;
    class observation;
  }
  user operation {
    uid 1002;
    class operation;
  }
  user engineering {
    uid 1003;
    class engineering;
  }
}
}

```

RADIUSの詳細な設定情報については、JUNOS設定ガイドをご覧ください。

ルーティングプロトコルセキュリティ

ルーターの本来の使命は、格納するルーティングテーブルや転送テーブルに基づいて、めざす宛先にユーザートラフィックを転送することです。このため、攻撃者がルーティングプロトコルの偽装パケットをルーターに送りつけて、ルーターのルーティングテーブルなどのデータベースの内容を改竄・破壊しようとする可能性があり、ルーターやネットワークの機能の低下を招きかねません。こうした攻撃を防ぐには、ルーティングプロトコルに基づく関係（ピアリングや隣接の関係）を確実に構築しなければなりません。その一例として、ルーティングプロトコルメッセージを認証する方法が挙げられます。ルーティングプロトコルの設定時には、認証を利用することを強くお勧めします。JUNOSソフトウェアは、BGP、OSPF、IS-IS、RIP、RSVPの各プロトコルでHMAC-MD5による認証をサポートしています。このHMAC-MD5は、送信するデータと組み合わせた秘密鍵を使ってハッシュを計算します。算出されたハッシュをデータとともに送信します。受信側では、対応する鍵を使って、メッセージハッシュを再計算・検証します。攻撃者がメッセージを偽装・改変した場合には、ハッシュが一致しないため、データは破棄されます。

次の例は、「internalpeers」という名称のBGPピアグループを対象とした単一鍵の設定方法です。また、BGP認証は、隣接ルーターやルーティングインスタンスのレベルで設定することも、全BGPセッションを対象にすることも可能です。どのようなセキュリティ設定にも共通して言えることですが、きめ細かさ（ある意味でセキュリティのレベル）を追求するほど、システム保守に伴う管理が煩雑になるというジレンマは避けられません。

```

protocols {
  bgp {
    group internalpeers {
      authentication-key XXXXXXXX;
    }
  }
}

```

JUNOSのIGPはすべて認証をサポートしていますが、IGPの中には本質的に特に安全性が高いものもあります。ほとんどのサービスプロバイダは、OSPFがIS-ISのいずれかを使って、高速な内部統合、拡張性、MPLSのトラフィックエンジニアリング機能の実現しています。OSPFは、IPでカプセル化されるために遠隔地からのスプーフィングやDoS攻撃に弱いのに対して、IS-ISは、ネットワーク層で利用できないため、スプーフィングしにくい特長があります。

ルーティングエンジンが受信するトラフィックのフィルタリング

前のセクションでは、信頼性におけるピアに限定してルーティング関係を構築する手段として、認証の利用を取り上げました。もちろん、これだけでもルーティングプロトコルの保護効果がありますが、ルーティングエンジン（RE）でのプロセスを悪意のあるパケットや不審なパケットから完全に守るには不十分です。たとえば、特定のプロトコルを標的に偽装パケットを送る方法で、ルーターを攻撃することが可能なのです。こうしたパケットは認証チェックではじかれませんが、それでもRE上でルーターのリソース（CPUサイクルや通信キュー）を消費させることになるため、結果的に攻撃は成功してしまうのです。このような事態を回避するには、信頼のおける送信元から送られたプロトコルパケットや制御パケットだけをREに到達させる仕組みが欠かせません。その仕組みとして、ジュニパーネットワークスでは、ファイアウォールフィルタの使用を推奨しています。ジュニパーネットワークス製ルーターは、全製品が転送速度への影響をなくすためハードウェアベースのファイアウォールを搭載しています。したがって、ファイアウォールフィルタを使用することで、性能を犠牲にすることなく、非常に強力で柔軟性に優れた保護体制が確立できます。

ルーティングエンジンのトラフィックにファイアウォールフィルタを適用

ルーティングエンジンを保護する場合、ファイアウォールフィルタは、ルーターのループバックインタフェースに適用するだけで済みます。ルーターの全インタフェースを対象にフィルタを追加・修正する必要はありません。このように、従来のルーターとは、手順が大きく異なります。

次の例は、入力フィルタとしてファイアウォールフィルタ「protect-RE」をlo0インタフェースに適用し、ルーティングエンジンを保護する設定を示しています。このフィルタ1つで、ユーザー側インタフェースからルーターに入ってくるルーティングエンジン宛の全トラフィックをチェックできます。なお、ファイアウォールフィルタ自体の設定については、次のセクションで取り上げます。

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        filter {
          input protect-RE;
        }
        address 10.10.5.1/32;
      }
    }
  }
}
```

ファイアウォールフィルタの設計

ルーティングエンジン宛トラフィックのうち、信頼のおけるトラフィックだけを許可するファイアウォールを作成する場合、次の事項を考慮に入れなければなりません。

- ルーターで実行する必要があるサービスやプロトコル
 - ルーティングプロトコル（BGP、OSPF、RSVPなど）
 - 管理サービス（SSH、DNS、NTPなど）
 - 診断・トラブルシューティング用プロトコル（ICMP、tracerouteなど）
- 上記のREでのサービスやプロトコルの宛先アドレスと宛先ポート
- 上記サービスに関して通信相手となるピアの信頼できる送信元アドレス
- 1で挙げた各サービスに割り当て可能なトラフィック速度

必要なサービスやプロトコルが決定すると、それぞれが信頼できる送信元アドレスと組み合わせられます。このアドレスから各サービスが送信され、ファイアウォール定義に適合条件が作成されます。これに対応するアクションとして、パケットを受け入れ、他のトラフィックを破棄します。

以下に挙げる設定例は、ルーティングエンジンを保護するファイアウォールに利用可能な3つのフィルタ条件を示しています。

```

firewall {
  filter protect-RE {
/* ポリサーの設定は省略 */
    term ssh {
      from {
        source-prefix-list {
          ssh-addresses;
        }
        protocol tcp;
        port [ ssh telnet ];
      }

      then {
        policer ssh-policer;
        accept;
      }
    }
    term bgp {
      from {
        source-prefix-list {
          bgp-sessions-addresses;
        }
        protocol tcp;
        port bgp;
      }
      then accept;
    }
  }
/* 信頼性が低い場合、特に警告せずに破棄し、分析用にログに記録します。 */
  term everything-else {
    then {
      syslog;
      log;
      discard;
    }
  }
}

```

ICMPフラッド攻撃とSYNフラッド攻撃への対処

ルーティングエンジンに対するICMPフラッド攻撃（ICMP ECHO攻撃）などの攻撃に対処するため、当社では、ルーター宛のICMPトラフィックにレートリミットを適用することをお勧めします。攻撃者は、数種類のICMPメッセージを利用して、ルーター機能を低下させたり、マシンのスキャンングを実行したりします。そこで、しかるべきネットワーク運用やトラブルシューティングに必要なICMPメッセージだけを許可する設定を推奨します。

TCPを利用したSYNフラッド攻撃も非常に一般的な攻撃方法で、攻撃者は、スクリプトやプログラムを使って、被害者側が解放するよりも速いスピードでTCPコネクション要求（SYNメッセージ）を実行します。このため、TCPのSYNメッセージに対してレートリミットを適用することを推奨します。TCPコネクションは、3ウェイハンドシェイクだけで確立できるため、受信するSYNパケットの速度を500Kbpsに制限すれば、安全に実行されます。

次に挙げるのは、SYNとICMPの速度を制限する2種類のファイアウォールフィルタ条件（例では、protect-REフィルタの場合）と、ICMPリダイレクトメッセージを拒否する条件です。

```

firewall {
  filter protect-RE {
    term police-init {

```

```

        from {
            source-prefix-list {
                ssh-addresses;
                bgp-addresses;
            }
            protocol tcp;
            tcp-flags "(syn & !ack) | fin | rst ";
        }
        then {
            policer tcp-init-policer;
            accept;
        }
    }

    term icmp {
        from {
            protocol icmp;
            icmp-type [ echo-request echo-reply unreachable time-
exceeded ];
        }
        then {
            policer small-policer;
            accept;
        }
    }
}

```

レートリミットを実行するポリサーの詳細なステートメントは、「ルーティングエンジンへのトラフィックのレートリミット」というセクションで解説します。SYN攻撃やICMP攻撃の詳細については、付録Aのリファレンスをご覧ください。

悪意あるフラグメンテーションへの対処

IPフラグメンテーションは、攻撃に悪用されることがあります。TCPパケットを偽装して、ルーターやホストに使われるIPフィルタをすり抜けさせることができます。RFC1858は、DoS攻撃に発展しかねない小さなオフセットの利用について解説したうえで、このような攻撃の対策には一般的なアルゴリズムの使用を推奨しています。JUNOSでは、このRFCで定義されているアルゴリズムを実装できます。

意外なこともかもしれませんが、多くのユーザーが採用しているファイアウォールの設定では、フラグメントが落ちています。次のフィルタを考えてみましょう。

```

term bgp{
    from {
        protocol tcp;
        destination-port bgp;
        source-address <trusted peer>;
    }
    then {
        accept;
    }
}

term default {
    then discard;
}

```

最初のパケットフラグメントには、宛先ポート（などの照合基準）が含まれているため、通過が許可されます。その後続く同一パケットのフラグメントには、宛先ポート情報が入っていないため、ファイアウォール条件に一致せずに破棄されます（デフォルト条件の場合）。

フラグメント化したパケット（BGPパケットなど）をRE宛に送りたい場合には、ファイアウォールフィルタlo0を次のように設定します。

```

term small-fragments {
  from {
    fragment-offset [1-5];
  }
  then {
    syslog;
    discard;
  }
}

term fragments {
  from {
    source-prefix-list {
      trusted-sources
    }
    is-fragment;
  }
  then {
    accept;
  }
}

```

「small-fragments」という条件は、小サイズのフラグメントをすべて破棄します。小サイズフラグメントは無効のため、破棄されます。2番目の条件で、信頼のおける送信元からのフラグメントはすべて受け入れることになります。

注：レイヤ4属性（TCPまたはUDPポート）のあるパケットを処理する前に、全フラグメントを処理する必要があります。

IPオプションを持つパケットへの対処

IPオプションは、特殊な状況で必要な制御機能を実現するものであり、実際の通信では、ほとんどの場合、必要ありません。オプションで実現可能な機能には、タイムスタンプや特殊ルーティングが挙げられます。このオプションは、自己宛でないパケットについても、入念に検査するよう中継ルーターに要求します。この結果、オプションが適用されたパケットは、ルーティングエンジンに送られて処理されます。このようなパケットが大量に押し寄せれば、ルーティングエンジンはお手上げとなり、完全なDoS（サービス拒否）状態に追い込まれます。バックボーンルーターがユーザートラフィックを運ぶだけのサービスプロバイダ環境では、バックボーンにしか存在しない制御トラフィックの一部を除けば、中継ルーターがIPオプション付きのパケットに遭遇することはありません。したがって、オプション付きパケットを落とすフィルタを導入して、ルーティングエンジンに到達させない対策が有効です。

サービスプロバイダがコアでMPLSを利用している場合、制御トラフィックに一定量のオプションが含まれることになります。こうした制御トラフィックとしては、「Router Alert」オプションを持つRSVPパケットが挙げられます。そこで、プロバイダによっては、オプション付きパケットを破棄せずに、レートリミットを適用することも可能です。

オプション付きパケットの宛先はルーター自体ではないため、ファイアウォールフィルタlo0では遮断できません。受信側インタフェースに推奨フィルタを適用することは可能ですが、このようなフィルタの管理は手間がかかります。特に、インタフェースやサブインタフェースが多数あるルーターの場合、この傾向が顕著に見られます。そこで、代替策として、ルーターの転送テーブルにフィルタを適用する方法があります。

制御トラフィックにはバースト性があるため、レートリミットではburst-size-limitの値が重要です。RSVPパケットのburst-size-limit値を決定するには、次の点を考慮に入れる必要があります。

1. RSVPパケットのサイズ
2. 1秒間にルーターが受け取るメッセージ数

第1の変数は、MPLSバックボーンで有効になるトラフィックエンジニアリングのオプションを利用します。RSVPパケットのサイズは、パケットサイズのログ記録と閲覧を直接実行することで確認できます。第2の変数は、ルーターを通るLSPの数によって決まります。中継ルーターの場合、受信したRSVPメッセージの大部分を保持しているため、ここではburst-size-limitの計算に中継ルーターのケースを想定します。RSVPパケット（IPヘッダ含む）は400バイトを設定します。中継ルーターは、両方向からRSVPメッセージを受け取る（双方向のLSPを想定）ため、最悪のケースを想定します。計算式は次のようになります。

400 bytes per packet * no of LSPs * 4

中継LSP数が200の場合、バーストサイズは320000バイトになります。

```

policer option-policer {
    if-exceeding {
        bandwidth-limit 3m;
        burst-size-limit 320000;
    }
    then discard;
}
    
```

そこで、オプション付きパケットのレートリミット用フィルタは、次のようになります。

```

filter filter-optioned {
    term one {
        from {
            ip-options any;
        }
        then {
            count option-packets;
            policer option-policer;
        }
    }
    term default {
        then accept;
    }
}
    
```

これは、転送オプションの設定階層下で転送テーブルに適用できます。

ルーターを攻撃のリフレクタに利用されないための対策

デフォルト設定の場合、ルーターは、経路不明のパケット送信元に対して、ICMPの「destination unreachable (到達不能)」というエラーメッセージを送り返します。この動作は、攻撃者がリフレクタとして悪用する恐れがあります。つまり、標的となるサーバー (被害者) から多数のルーターに要求があったかのように攻撃者が要求を偽装すると、今度は、要求を受け取った他のルーターからの応答がまとまって標的に戻ってきます。攻撃者は、送信元アドレスに標的となるアドレスを設定したうえで、存在しない宛先にパケットを送ります。すると、大量のICMPエラーメッセージが標的に戻されるのです。ルーターをリフレクタに利用させないためには、宛先が存在しないパケットは、無条件に破棄するのも手です。また、「discard」(破棄)のインタフェースをデフォルトの経路に設定しておくのも、リフレクション攻撃対策として有効です。こうすれば、転送テーブルに有効な経路がないパケットは、自動的に破棄されます。さらに、こうした破棄をログに記録するフィルタをインストールしておけば、ネットワーク管理者が攻撃のトラフィックを分析することも可能です。設定例を以下に示します。

最初に、プライベートIPアドレスを使って「discard」のインタフェースを設定します。

```

dsc {
    unit 0 {
        family inet {
            filter {
                input log-discard;
            }
            address 10.1.1.1/32 {
                destination 10.1.1.2;
            }
        }
    }
}
    
```

「discard」インタフェースにフィルタを設定し、破棄対象のパケットを監視します。

```

filter log-discard {
    term one {
        then {
            syslog;
            discard;
        }
    }
}

```

デフォルトの経路を設定し、
転送テーブルに経路が存在しないパケットを破棄用インタフェースに送ります。

```

static {
    route 0.0.0.0/0 next-hop 10.1.1.2 ;
}

```

VPNを巡る課題

VPNを使うと、サービスプロバイダがユーザーに私設網のメリットを提供できます。しかも、IPインフラは共用のため、コスト削減のメリットもあります。VPNは、専用の転送テーブルを利用し、あたかも私設網のように扱えます。ユーザーごとに独立した転送機能に、MPLSの伝送を組み合わせることで、IPバックボーンを通るトラフィックを明確に安全に分離できます。しかし、こうした転送テーブルの作成・管理は、共通の制御プレーンで実行します。つまり、制御プレーンとリソースを共用するため、機能低下や不正なピアからVPNを守るため、手厚い保護対策が欠かせません。

今回推奨する方法は、仮想ネットワークごとに適用し、CE（カスタマーエッジ）機器かPE（プロバイダエッジ）機器のいずれかが対象となります。

たとえば、VPN内に攻撃の影響を受けたホストが1台でもあると、このホストがPEの制御プレーンに攻撃を仕掛ける恐れがあります。その結果、同じPEを共有する他のVPNにも影響が及びかねません。挙動が怪しいVPNに対処するためには、さまざまな安全策が考えられます。

PEとCEの間での経路配布には、BGPやOSPF、RIPなど、さまざまなルーティングプロトコルが使われます。BGPとRIPはそれぞれ指定したレイヤ4ポートを使用して通信するため、ホストを利用して、ネットワーク外部からポートに大量のパケットを簡単に送りつけ、DoS攻撃を実行できます。一方、OSPFは、レイヤ3で動作するため、IPのペイロードとして情報をやり取りします。したがって、攻撃の影響を受けたホストがOSPFパケットを偽装しようとする場合、PE-CE間のリンクでローソケットサービスへのアクセス権が必要になります。つまり、非常に面倒な手順を踏まなければ攻撃できません。そこで、PE-CE間のルーティングプロトコルを選定する際には、OSPFの検討をお勧めします。

有効なルーティング更新情報がPEデバイスに届くためには、対応するCEルーターから情報を送る必要があります。そこで、社内LANに対向するCE側インタフェースに、受信用のファイアウォールフィルタを設置し、PEデバイス宛のルーティング更新情報や制御更新情報を1つ残らず破棄する方法も可能です。

```

filter protect-PE {
    term one {
        from {
            destination-address {
                192.1.1.1/32;      # PE's address
            }
            protocol bgp;          # Routing protocol between PE and
a CE
        }
        then {
            log;
            discard;
        }
    }
}

```

各VPNでは、閉域網単位で一定レベルの仮想化を実現します。つまり、各ユーザーが専用のルーティングテーブルとルーティングインスタンスを持っています。そこで、リソースの乱用を防止するため、閉域網ごとに利用可能なリソースの上限を設定しておくことを強くお勧めします。

こうしたリソースの例としては、ルーティングテーブルのメモリが挙げられます。各VPNでは、PEルーター上に専用のルーティングテーブル（転送テーブル）を持っています。CEルーターがPEに送った経路情報は、適切なルーティングテーブル（転送テーブル）に反映されます。このため、挙動の怪しいCEルーターがPEに予定以上の量の経路情報を送れば、PEのメモリが消費され、場合によっては別のVPNの経路情報用に予約されているメモリまで使ってしまうこともあります。このような事態は、次の設定で回避できます。

```
[edit routing-instances]
vpn {
  routing-options {
    maximum-routes {
      100;
      threshold 80;
    }
  }
}
```

各VPNにmaximum-routesを設定すると、特定のVPNがPEルーターのメモリを必要以上に消費してしまうことを防止できます。thresholdの値は、警告メッセージが発行される上限（%）を表します。

かなりひどい状況になると、不正なVPNがPEルーターへのルーティング更新情報を大量に送りつけ、他のVPNからCPUパワーを奪ってしまうこともあります。そこで、CEルーターからのプロトコル更新情報にレートリミットを適用すれば、この問題に対処できます。

PE宛の他の制御トラフィックは、そのPE自身が破棄するのが理想です。しかし、プロバイダとしては、他の制御トラフィック（トラブルシューティング用のICMPトラフィックなど）を排除するわけにはいかない場合もあります。こうしたプロトコルのリスクは小さいため、ポリシー適用やレートリミットで適正な値に制限する方法が使えます。

各VPNへのファイアウォールフィルタの適用

VPNごとに定義したフィルタを、PEルーターのメインのループバックインタフェースに適用するファイアウォールフィルタにまとめて埋め込むこともできます。しかし、論理的なループバックフィルタをVRFごとに定義する方法であれば、各VRFが識別しやすくなります。これは、特にトラブルシューティングに有効な方法で、ユーザーはリモートのCEにpingを実行してローカルPEルーターの情報を確保できます（この機能の詳細については、JUNOSソフトウェア設定マニュアルをご覧ください）。なお、この方法には、次のようなメリットもあります。

ファイアウォールフィルタが管理しやすくなります。新規のVPNを追加する際、対応するループバックインタフェースに、独立したファイアウォールフィルタが適用されます。

送信元が信頼のおけるCEやPEではないパケットは、すべて各VPNのフィルタが破棄するため、インターネット経由でVPNに攻撃を仕掛けることが困難になります。

ルーティングエンジンへのトラフィックのレートリミット

ルーティングエンジンの保護に当たっては、許可されている各サービスのトラフィック負荷を一定レベルに抑えることが大切です。そのレベルとは、前出のフィルタリング（P9）で取り上げた分析によって決まります。制御トラフィックの速度を制限することで、正当なトラフィックを装った偽装パケットを高速で送りつけるDoS攻撃からルーティングエンジンを保護できます。

ルーターを正しく機能させるうえで、ルーティングと制御のトラフィックは欠かせません。このため、ネットワークが不安定な状態のときに、ネットワークを安定させるには、ルーティングプロトコルの迅速な統合が重要です。ルーティングプロトコルのトラフィック量を制限してさまざまな攻撃に対処するのが好ましいように思えますが、プロトコルトラフィックの最大速度を一律に決定することはきわめて困難です。というのも、最大速度は、ピアや隣接機器の数に左右されるうえ、時間の経過とともに変化します。したがって、ルーティングプロトコルのトラフィックにはレートリミットを適用しないのが最善策と言えます。

一方、管理トラフィックは、ルーティングトラフィックに比べて、重要性が低く、速度もある程度見当がつけやすいため、一定の速度に設定することで、柔軟性の低いトラフィックにリソースが消費されるのを防ぐことができます。当社では、管理トラフ

ティックの種類別に一定の帯域を割り当てることをお勧めします。こうしておけば、何らかのサービスを利用して攻撃を仕掛けられても、ルーターのCPUパワーが使い切られてしまうことはありません。

こうした攻撃の防止に効果的な値を使ったポリサー設定を以下に示します。

```

firewall {
  filter protect-RE {
    policer ssh-police {
      if-exceeding {
        bandwidth-limit 1m;
        burst-size-limit 15k;
      }
      then discard;
    }
  }
  policer tcp-init-policer {
    if-exceeding {
      bandwidth-limit 500k;
      burst-size-limit 15k;
    }
    then discard;
  }
}
/* ファイアウォールの条件は省略* /
  
```

付録Bでは、このほかのポリサー設定例や応用例が紹介されています。

セキュリティのための監査

ネットワーク事業者は、自営網で重大な問題が発生した場合には、事態の推移を監視する責任があります。その際、主たる情報源は、通常、システムイベントのログです。イベントやアクションをログに記録しても、それだけでルーターのセキュリティが強化されるわけではありませんが、現行のセキュリティポリシーやルーター設定の有効性を判断するうえでログが役に立ちます。また、継続的で計画的な攻撃への対策を考えるときにも、ログを使えば、攻撃に使われたトラフィックの送信元アドレスやルーター、ポートの特定に役立ちます。さらに、攻撃者を告訴する際には、ログが法律的に重要な証拠となります。プロトコルやルーターの重要イベントは、もれなくリモート環境のシステムログ (syslog) サーバーに記録しておくことをお勧めします。記録されたsyslogファイルをリアルタイムに解析して、不審な動きを見つけたしたり、いざというときの資料としてアーカイブ化しておいたりすることも可能です。

ルーターやプロトコルのイベントは多数あるため、ログの対象とするイベントを決めなければなりません。その際には、数多くの要素が関わってくるため、ネットワークごとに条件は違ってきます。ただし、一般的に、イベントは、以下の3つのカテゴリーに大別できます。

- 認証・コマンドのイベント
- ルーティングプロトコルのイベントとエラー
- 拒否されたトラフィック

認証・コマンドのイベントログ作成

ルーターの管理アクティビティを完璧に追跡するには、認証・権限付与や拒否、ユーザーコマンドの実行などをファイルに記録しておくのが理想的です。認証失敗のイベントが記録されたファイルをチェックすれば、ルーターに対するハッキングの動きを突き止めることができます。また、ファイルには、ルーターで実行された全コマンドと実行者がログに記録されています。ルーターで実行されたコマンドのログを調べ、特定の時間に実行された変更とネットワークでのイベントを突き合わせていくことも可能です。

ルーティングプロトコルのイベントとエラー

ルーティングプロトコルのイベントやエラーは、ルーティングプロトコルに対する攻撃のバロメーターです。たとえば、イベントには、プロトコル認証の失敗も含まれます。つまり、何らかの挙動を引き起こそうと、偽装や異常があるルーティングパケットをルーターに送ろうとしている攻撃者を割り出すこともできます。

拒否されたトラフィックのログ作成

前出のフィルタリング (P9) で述べたように、攻撃者を上手に特定するには、ルーターのファイアウォールフィルタを通過できなかったトラフィックをもれなくログに記録しておくことが肝心です。syslogファイルをリアルタイムに解析するスクリプトを用意し、詳細な調査が必要な場合に指定したエンジニアにアラームが送られるようにしたり、後で精査できるようにログを保管しておいたりすることも可能です。このようにログを取っておくと、ルーターに対する攻撃の有無、発生時刻はもちろん、攻撃に使われたパケットの送信元やタイプまで調べることもできます。アクティブユーザーに緊急情報を送り、2種類のsyslogサーバーに別々のsyslog情報を送る設定を以下に示します。

```
syslog {
    user * {
        any emergency;
    }
    host 10.1.3.1 {
        authorization any;
        daemon info;
        kernel notice;
        interactive-commands any;
    }
    host 10.1.3.2 {
        authorization any;
        daemon info;
        kernel notice;
        user notice;
        interactive-commands any;
    }
}
```

NTP (Network Time Protocol)

全ルーターのログファイルにあるタイムスタンプの同期が取れていれば、デバッグやトラブルシューティングの作業が非常に楽になります。たとえば、ネットワーク全体に及ぶイベントを複数のログに同時に記録された内容と突き合わせて、対応関係を見つけることができます。そこで、ルーターなどのネットワーク機器のシステムクロックの同期には、NTP (Network Time Protocol) の使用を強くお勧めします。

NTPは、デフォルトでは、まったく認証を受けない形で動作します。ルーターのクロックの精度を狂わせようとする悪意ある行為があれば、システムロギングに影響が及びかねません。その結果、トラブルシューティングや侵入検知は困難になり、他の管理機能の妨げにもなります。

そこで、認証を有効にした典型的なNTP設定をご紹介します。

```
ntp {
    authentication-key 2 type md5 value "XXXXXXXXXXXX"; # SECRET-
    DATA
    boot-server 10.1.4.1;
    server 10.1.4.2;
}
```

boot-serverのステートメントでは、ルーター起動時に最初に曜日と日付けを取得するサーバーを指定します。

serverステートメントでは、定期的な時刻同期に使用するNTPサーバーを指定します。authentication-keyステートメントでは、認証の鍵値のハッシュにHMAC-MD5方式が使用されることを指定します。こうすることで、ルーターがタイムサーバーを装った攻撃者のホストに同期してしまう事態を回避できます。

まとめ

効果的なセキュリティ計画を実施する場合、使いやすさを取るか、それとも鉄壁さを取るかという難しい選択を迫られます。今日のネットワークサービスプロバイダの環境では、セキュリティ確立に当たって、管理の簡素化をめざしつつも、攻撃者から見て強固な体制づくりを追求しなければならず、最終的に妥協点を探る必要があります。本資料で取り上げたツールやテクニックは、ネットワークインフラとエンドユーザーの双方のニーズに応えるセキュリティポリシーの実施に役立ちます。

セキュリティポリシーと、これに基づく設定は、一度決めたらそれで終わりという性格のものではなく、時間の経過とともに進化しなければなりません。攻撃者がネットワークインフラ侵入の新しい手口を編み出せば、自社にとってのネットワークの「安全」基準を見直し、状況に合った設定に変更する必要があります。ネットワークの脅威そのものとは直接関係のない問題であっても、こうした見直しが必要な場合があります。たとえば、特定タイプの攻撃からの保護を重視するユーザーを新規に獲得した場合や、新規のサービスやプロトコルを導入した場合なども、これまでのセキュリティ環境を見直すきっかけになります。

そこで、ネットワーク要件の観点から本資料で取り上げた各事項について検討し、ニーズに合ったものを導入するだけでなく、定期的にセキュリティポリシーをチェックし、ネットワークに対する新たな脅威に対応できているかどうかを確認することをお勧めします。こうした定期的な見直し作業に、オペレーティングシステムの更新作業も連動させた取り組み??。“ニューパブリックネットワーク”のセキュリティ課題に対処できる万全のルーター環境づくりは、ここから始まります。

付録A

DoS攻撃（DDoS攻撃を含む）の歴史や技術情報などをまとめた参考サイトをいくつかご紹介します。

- <http://rr.sans.org/threats/DDoS.php>
- <http://rr.sans.org/securitybasics/dos.php>
- http://www.juniper.net/techcenter/app_note/350001.html

付録B

```
system {
```

```
    host-name    Secure-Router;
```

```
    domain-name  company.com;
```

```
    default-address-selection;
```

/* 認証方式は、RADIUSのみに設定されているため、RADIUSポリシーが実行されます。RADIUSサーバーがアクセス不能の場合、ローカルアカウントが使用されます。*/

```
    authentication-order [ radius ];
```

```
    root-authentication {
```

```
        encrypted-password " XXXXXXXXXXXXXXXXXXXXXXXX"; # SECRET-DATA
```

```
    }
```

```
    name-server {
```

```
        10.1.1.1;
```

```
        10.1.1.2;
```

```
}
```

/* RADIUS認証を有効にして、クライアントとサーバーの間で共有する秘密情報を定義することで、お互いに通信相手が信頼のおけるピアであると確認できます。タイムアウトを設定すれば、一定秒数が経過後にバックアップサーバーや別の認証方式に切り替えることができます。*/

```

radius-server {
    10.1.2.1 {
        secret "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"; # SECRET-DATA
        timeout 5;
    }
    10.1.2.2 {
        secret "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"; # SECRET-DATA
        timeout 5;
    }
}

/* 複数のログインアカウントを作成し、それぞれに個別の特権を設定します。タイムアウトを設定し、ユーザーのアカウント、
のアクティビティがなくなってから一定秒数が経過後に切断するのもいいでしょう。ユーザーの特権レベルは、組織内での職
務・任務を反映したものとします。 */login { /* このユーザークラスは、統計情報と設定しか閲覧できません。設定を修正する、
権限はありません。 */

login {

/* このユーザークラスは、統計情報と設定しか閲覧できません。設定を修正する権限はありません。 */

    class provisioning {
        idle-timeout 5;

        permissions [ configure firewall interface network routing snmp
system trace view];
    }

/* このユーザークラスは、設定の閲覧と修正が可能です。 */

    class operation {
        idle-timeout 5;

        permissions [admin clear configure interface interface-control netwo
reset routing routing-control snmp snmp-control trace-control firewall-contr
rollback];
    }

/* このユーザークラスは、アクセス権と制御権に制約がありません。 */

    class engineering {
        idle-timeout 5;

        permissions all;
    }
}

```

/* これはローカルのスーパーユーザーのアカウントです。RADIUSの障害時やサーバーのアクセス不能時には、ルーターのローカルログインアカウントを使用してください。 */

```
user admin {  
    uid 1000;  
    class engineering;  
    authentication {  
        encrypted-password "<PASSWORD>"; # SECRET-DATA  
    }  
}
```

/*ユーザー単位またはユーザーグループ単位でRADIUSテンプレートを定義します。*/

```
user observation {  
    uid 1001;  
    class observation;  
}  
user operation {  
    uid 1002;  
    class operation;  
}  
user engineering {  
    uid 1003;  
    class engineering;  
}  
}
```

/* ルーターの接続サービスを有効にします。

対象ルーターで可能な同時接続数と1分当たりの接続試行回数を設定します。 */

```
services {  
    ssh connection-limit 10 rate-limit 4;  
}
```

/* syslogファイルにどのメッセージを登録し、どのメッセージをリモートサーバーに送るのかを指定します。ユーザーが syslogサーバーごとに別のメッセージを送ることもできます。また、緊急メッセージの場合、ログイン中の全ユーザーに送信することも可能です。なお、複数のsyslogサーバーを用意しておく、万一の障害時でも安心です。*/

```

syslog {
    user * {
        any emergency;
    }
    host 10.1.3.1 {
        authorization any;
        daemon info;
        kernel notice;
        interactive-commands any;
    }
    host 10.1.3.2 {
        authorization any;
        daemon info;
        kernel notice;
        user notice;
        interactive-commands any;
    }
}

/* syslogファイルは手元にも保存しておくことをお勧めします。*/

file messages {
    any notice;
    authorization info;
    daemon any;
    kernel any;
    archive size 10m files 5 no-world-readable;
}

file authorization-commands {
    authorization any;
}

```

```
        interactive-commands any;
    }
/* ファイアウォールログは独立したsyslogファイルに記録されます。*/

    file firewall-logs {
        firewall any;
    }
}

/* ネットワーク内の全ルーターを共通のタイムソースに同期させます。NTPのピアの信頼性が確認できるように、認証の利用をお勧めします。*/

ntp {
    authentication-key 2 type md5 value "XXXXXXXXXXXXXXXX"; # SECRET-DATA
    boot-server 10.1.4.1;
    server 10.1.4.2;
}

interfaces {
    at-4/0/0 {
        description core router;
        atm-options {
            vpi 0 maximum-vc 1024;
            ilmi;
        }
        unit 131 {
            description to-other-core-router;
            encapsulation atm-snap;
            point-to-point;
            vci 0.131;
            family inet {
                address 12.1.1.1/30;
            }
        }
    }
}
```

```

    }

    family iso;
}

```

/* 管理用のイーサネットインタフェースは、帯域外管理に利用できます。しかし、ほとんどのサービスプロバイダは、管理用として（運用コストが安上がり）帯域内通信を利用しているため、当社ではこのインタフェースを使用不可にして、ルーターのセキュリティ強化に取り組んでいます。*/

```

fxp0 {
    disable;
}

lo0 {
    unit 0 {
        family inet {
            filter {

```

/* このフィルタは、ルーティングエンジンへのトラフィックの受け入れと拒否を状況に応じて選択します。ルーターのループバックインタフェースだけに適用します。*/

```

                input protect-RE;
            }

            address 10.10.5.1/32;
        }

        family iso {
            address 48.0005.80dd.f900.0000.0001.0001.0000.0000.011.00;
        }
    }
}

so-2/0/0 {
    description To-other-router;

    clocking external;

    sonet-options {
        fcs 32;

        payload-scrambler;

```

```
    unit 0 {  
        family inet {  
            address 10.1.5.1/30;  
        }  
        family iso;  
    }  
}
```

/* ここで「com1」と定義したコミュニティは、一例にすぎません。ハッカーの悪用を防止するため、権限付与は読み出しのみに設定することをお勧めします。*/

```
snmp {  
    community com1 {  
        authorization read-only;  
    }  
    trap-group jnx-traps {  
        version v1;  
        targets {  
            10.1.6.1;  
        }  
    }  
}  
routing-options {  
    router-id 10.1.7.1;  
    autonomous-system 222;
```

/* RFC1918に規定されているプライベートアドレス空間は、ブロックすることをお勧めします。このようなアドレスや実体のない偽アドレスは、デフォルトのmartianアドレスに追加できます。*/

```
martians {  
    1.0.0.0/8 exact;
```

```

10.0.0.0/8 exact;
19.255.0.0/16 exact;
59.0.0.0/8 exact;
129.156.0.0/16 exact;
172.16.0.0/12 exact;
192.0.2.0/24 exact;
192.5.0.0/24 exact;
192.9.200.0/24 exact;
192.9.99.0/24 exact;
192.168.0.0/16 exact;
224.0.0.0/3 exact;
}
}
protocols {
    bgp {
        group ibgp {
            type internal;
            traceoptions {
                file bgp-trace size 1m files 10;
                flag state;
                flag general;
            }
            local-address 10.10.5.1;
            log-updown;
            neighbor 10.2.1.1;
                authentication-key "XXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        }
        group ebgp {
            type external;

```

```
        traceoptions {
            file ebgp-trace size 10m files 10;

            flag state;

            flag general;
        }
        local-address 10.10.5.1;
log-updown;

        peer-as 2;

        neighbor 10.2.1.2;
            authentication-key "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        }
    }
isis {
    authentication-key "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"; # SECRET-DATA
    authentication-type md5;

    traceoptions {
        file isis-trace size 10m files 10;

        flag normal;

        flag error;
    }

    interface at-0/0/0.131 {
        lsp-interval 50;
        level 2 disable;
        level 1 {
            metric 3;

            hello-interval 5;

            hold-time 60;
        }
    }
}
```

```

interface lo0.0 {
    passive;
}

```

/* 以下のアドレスは、プロトコルやサービスの通信相手として、信頼のおける送信元アドレスと考えられます。設定しやすさと読みやすさを考え、ここではプリフィックスリストを定義しておき、詳細はフィルタ定義を参照する方式にしました。また、以下の例では、プライベートアドレス空間を利用している点にご注意ください。プライベートアドレス空間を利用したい場合、martianアドレスリストから該当アドレスを除去しておく必要があります。*/

```

prefix-list ssh-addresses {
    192.168.122/24
}

prefix-list bgp-addresses {
    10.2.1.0/24;
}

prefix-list ntp-addresses {
    10.1.4.0/24
}

prefix-list snmp-addresses {
    10.1.6.0/24;
}

prefix-list dns-address {
    10.1.1.0/24;
}

prefix-list radius-address {
    10.1.2.0/24;
}

```

/* ルーティングエンジンを保護するフィルタです。*/

```

firewall {
    filter protect-RE {

```

/* サービスごとに一定の帯域または速度を割り当てておくことで、攻撃者がフィルタを突破（送信元アドレスの偽装など）するようになることになっても、プロトコルトラフィックが保護されます。*/

```
    policer ssh-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 15k;
        }
        then discard;
    }

    policer small-bw-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 15k;
        }
        then discard;
    }

    policer snmp-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 15k;
        }
        then discard;
    }

    policer ntp-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 15k;
        }
        then discard;
    }
```

```
    }  
  
    policer dns-policer {  
        if-exceeding {  
            bandwidth-limit 1m;  
            burst-size-limit 15k;  
        }  
        then discard;  
    }  
  
    policer radius-policer {  
        if-exceeding {  
            bandwidth-limit 1m;  
            burst-size-limit 15k;  
        }  
        then discard;  
    }  
  
    policer tcp-policer {  
        if-exceeding {  
            bandwidth-limit 500k;  
            burst-size-limit 15k;  
        }  
        then discard;  
    }  
}
```

/* 以下の条件では、信頼のおける送信元からのトラフィックだけを受け入れます。信頼のおけるトラフィックは、ルーティングプロトコルを除き、レートリミットの対象となります。詳しい説明については、本資料の「ファイアウォールフィルタの設計」セクションをご覧ください。*/

```
    term icmp {  
        from {  
            protocol icmp;  
            icmp-type [ echo-request echo-reply unreachable time-  
exceeded ];
```

```
    }
  then {
    policer small-bw-policer;
    accept;
  }
}

term tcp-connection {
  from {
    source-prefix-list {
                                ssh-addresses;
                                bgp-addresses;
    }
    protocol tcp;
    tcp-flags "(syn & !ack) | fin | rst";
  }
  then {
    policer tcp-policer;
    accept;
  }
term ssh {
  from {
    source-prefix-list {
                                ssh-addresses;
    }
    protocol tcp;
    port [ ssh ];
  }
  policer ssh-policer;
  then accept;
}
```

```

}
term bgp {
    from {
        source-prefix-list {
                                                    bgp-sessions-addresses;
        }
        protocol tcp;
        port bgp;
    }
    then accept;
}
term snmp {
    from {
        source-prefix-list {
                                                    snmp-addresses;
        }
        protocol udp;
        port snmp;
    }
    then {
        policer snmp-policer;
        accept;
    }
}
term ntp {
    from {
        source-prefix-list {
                                                    ntp-addresses;
        }
    }
}

```

```
        protocol udp;

        port ntp;

    }

    then {

        policer ntp-policer;

        accept;

    }

}

term dns {

    from {

        source-prefix-list {

            dns-addresses;

        }

        protocol udp;

        port domain;

    }

    then {

        policer dns-policer;

        accept;

    }

}

term radius {

    from {

        source-address {

            radius-addresses;

        }

        protocol udp;

    }

}
```

```

        port radius;
    }
    then {
        policer radius-policer;
        accept;
    }
}

term trace-route {
    from {
        protocol udp;
        destination-port 33434-33523;
    }
    then {
        policer small-bw-policer;
        accept;
    }
}

/* 信頼のおけないトラフィックはすべて警告なしに破棄されます。後で分析できるように、拒否されたトラフィックをログに記録しておくことをお勧めします。*/

term everything-else {
    then {
        syslog;
        log;
        discard;
    }
}
}

```

ジュニパーネットワークスの製品とサービス

詳細情報については、以下までお問い合わせください。

ジュニパーネットワークス株式会社
〒163-1035 東京都新宿区西新宿3-7-1
新宿パークタワー N棟 35階
電話 03-5321-2600
FAX 03-5321-2700
URL <http://www.juniper.co.jp>

Copyright © 2003, Juniper Networks, Inc. All rights reserved.

Juniper Networks は米国特許庁に登録されています。また、Juniper Networks は諸外国においてJuniper Networks Inc. の商標として登録されています。Broadband Cable Processor, ERX, ESP, E-series, G1, G10, G-series, Internet Processor, Juniper Your Net, JUNOS, JUNOScript, M5, M10, M20, M40, M40e, M160, M-series, NMC-RX, SDX, ServiceGuard, T320, T640, T-series, UMC, Unison はJuniper Networks Inc. の商標です。その他記載されている商標、サービスマーク、登録商標、登録サービスマークは各所有者に所有権があります。これらの仕様は予告なく変更されることがあります。ジュニパーネットワークスは、記載内容に誤りがあった場合でも、その責任は負いません。ジュニパーネットワークスは、予告なく本発行物を変更、修正、転載、または改訂する権利を持っています。