

PBL3: 演習

東京大学
明石邦夫

演習

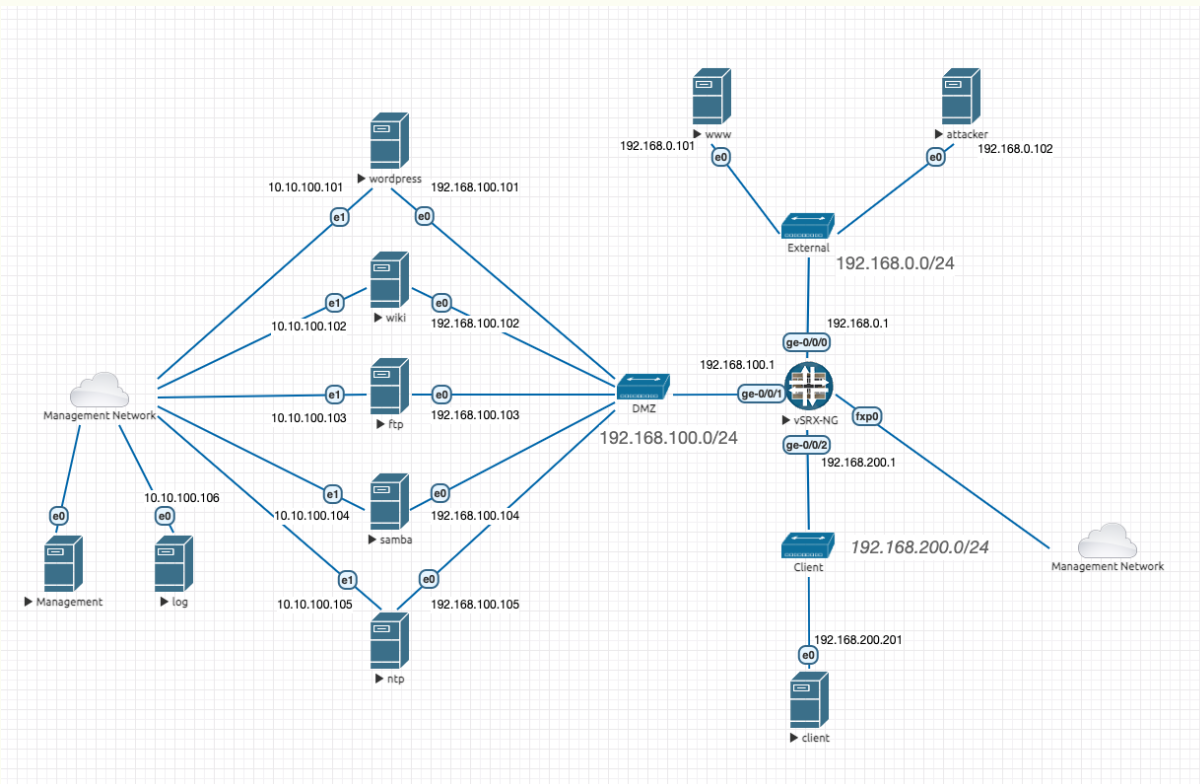
演習環境

- EVE-NG
 - ネットワーク機器を動作させるエミュレーション基盤
 - Cisco、Juniper、F5 など多数の機器が扱える
 - 仮想アプライアンスのイメージファイルは自分で用意
- 今回は Juniper SRX の仮想版である vSRX を使用

演習内容

External、DMZ、Client ネットワークがある環境で、Firewall の設定を行う

また、各サービスに対して適切なポリシーを設定し、それが正しいことを確認する



演習環境へのアクセス

- https と ssh でアクセス可能
 - どちらも Boundary から接続する必要あり

EVE-NG の使い方

- ブラウザでアクセス
- ログイン画面
 - ユーザ名: admin
 - パスワード: eve
- 3つ目は HTML5 console




演習環境の操作方法

- ログイン後、ネットワーク図が表示される

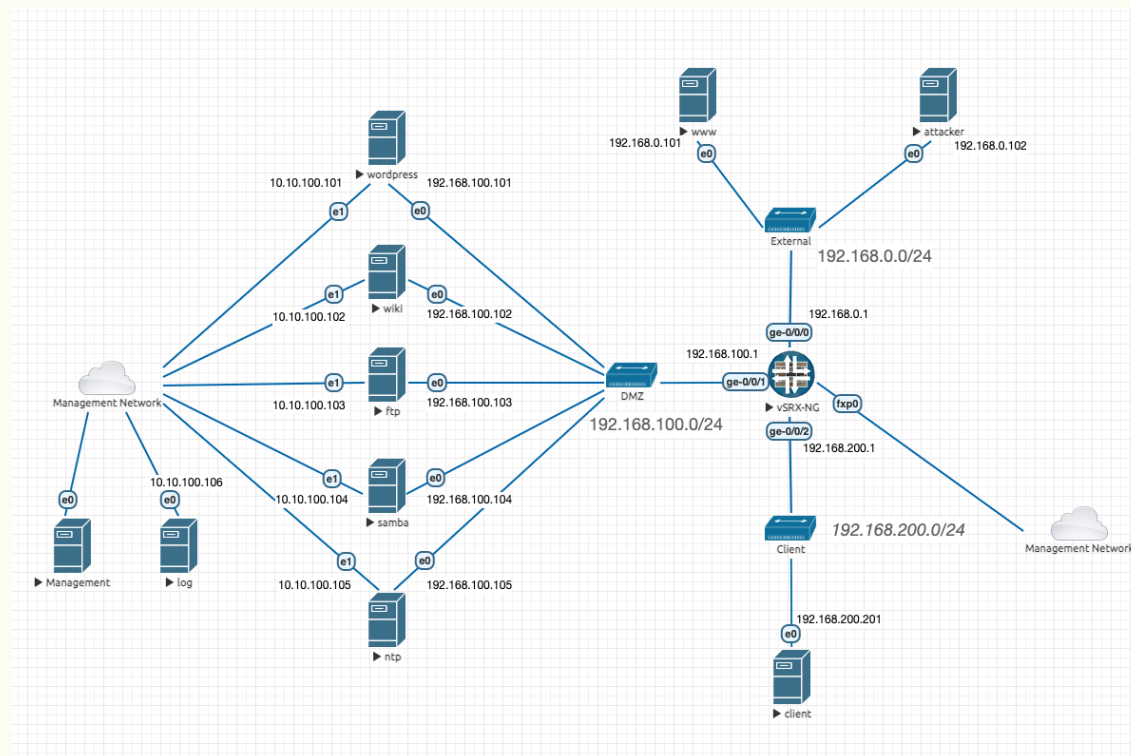
-  はFirewall のアイコン

-  はサーバ/クライアントのアイコン

-  はスイッチのアイコン

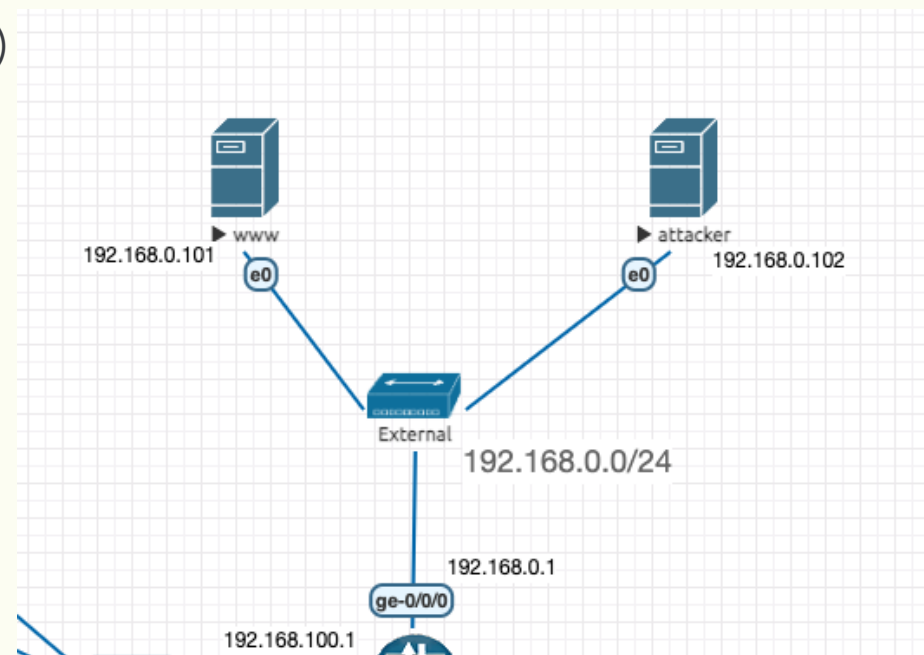
- 今回は操作不可

- Firewall、サーバ/クライアントは
起動していればクリックすると画面が切り替わる



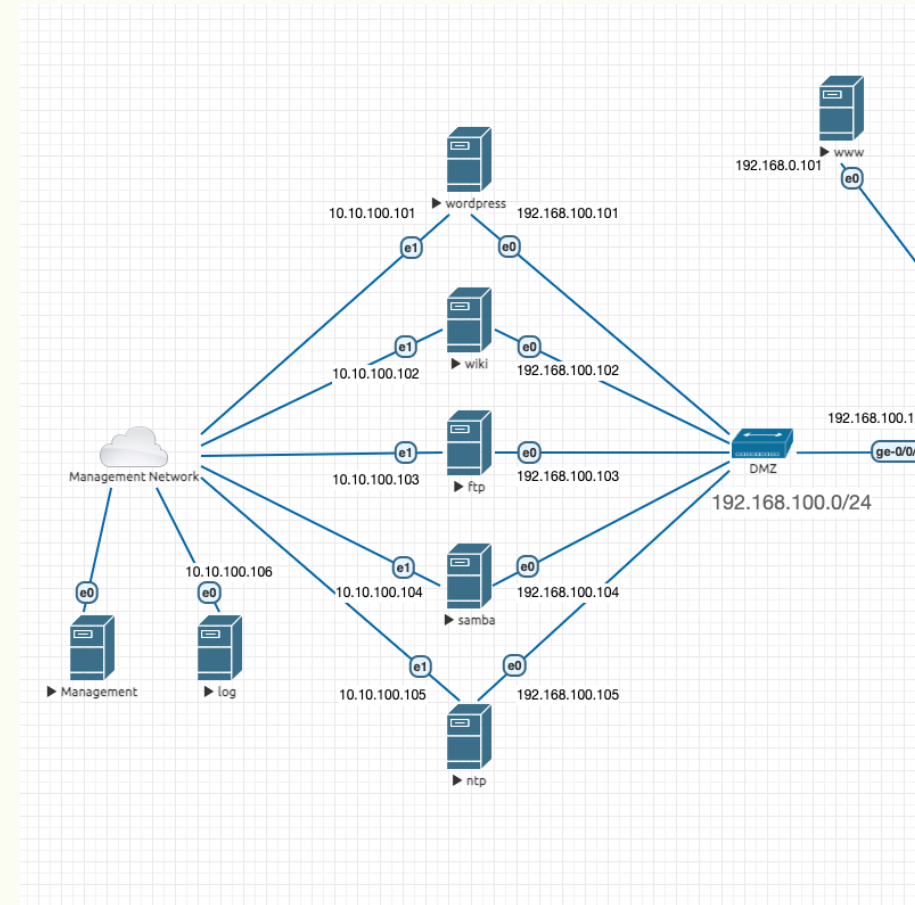
演習環境のネットワーク

- External
 - インターネット上のサーバ、クライアント(の想定)
 - 外部からの到達性
 - 攻撃された場合に防御できているか
 - 外部公開しているサービスにアクセス可能か
 - を確認するために使用



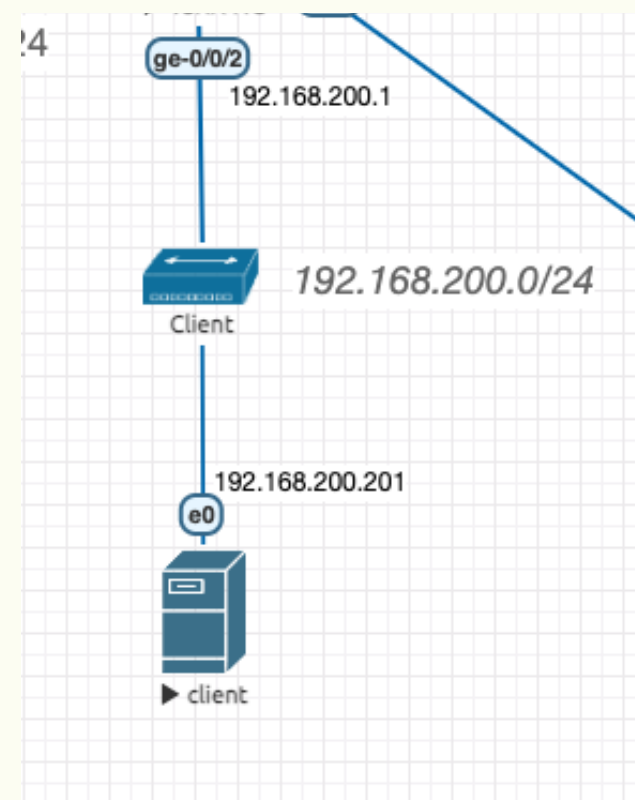
演習環境のネットワーク

- DMZ
 - 社内サービスのネットワーク(の想定)
 - http、ftp、ntp などのサーバが動作
 - 内部専用、外部に公開するサービスが混在
 - 内部専用サービスは内部からのみ
外部に公開しているサービスは外部からも
アクセスできるかを確認するために使用
- DMZ のサーバはすべて英語キーボードとなっている
 - 日本語キーボードの場合、記号が打てない場合がある
 - 後述する ssh でログインして操作すること



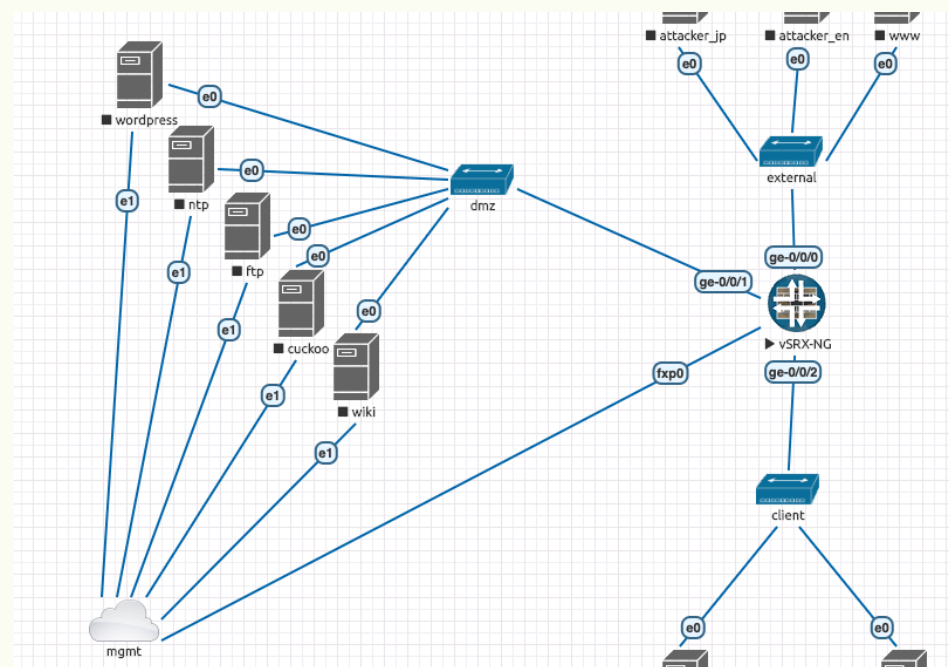
演習環境のネットワーク

- client
 - 社内のクライアント(の想定)
 - 外部への到達性
 - 正しく外部のサービスにアクセスできるか
 - 社内のサービスにアクセスできるか
 - 外部からクライアントに攻撃されないか
 - を確認するために使用



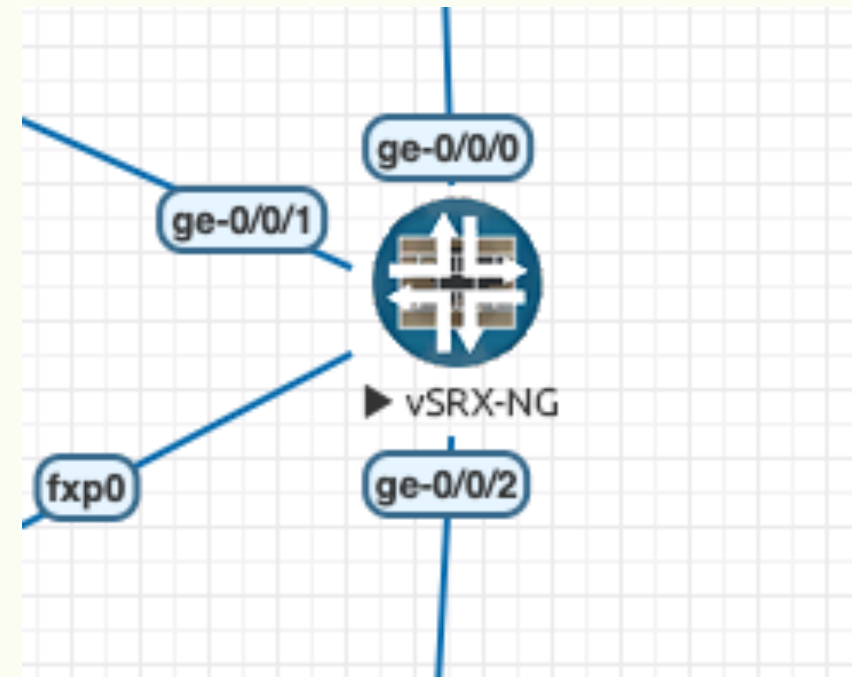
演習環境のネットワーク

- mgmt
 - 演習環境の管理ネットワーク
 - enpit-gw.jais.co に ssh するとアクセス可能
 - 踏み台からログインして操作
 - 各サーバへは ssh
 - SRX へは telnet/ssh が可能
- mgmt のアドレスは
 - 100.64.X.Y/24
 - X はアカウント名の pXXX の XXX
 - Y は DHCP でアドレスを取得するようサーバに設定済み
 - 設定されるアドレスは別途記載



Juniper SRX

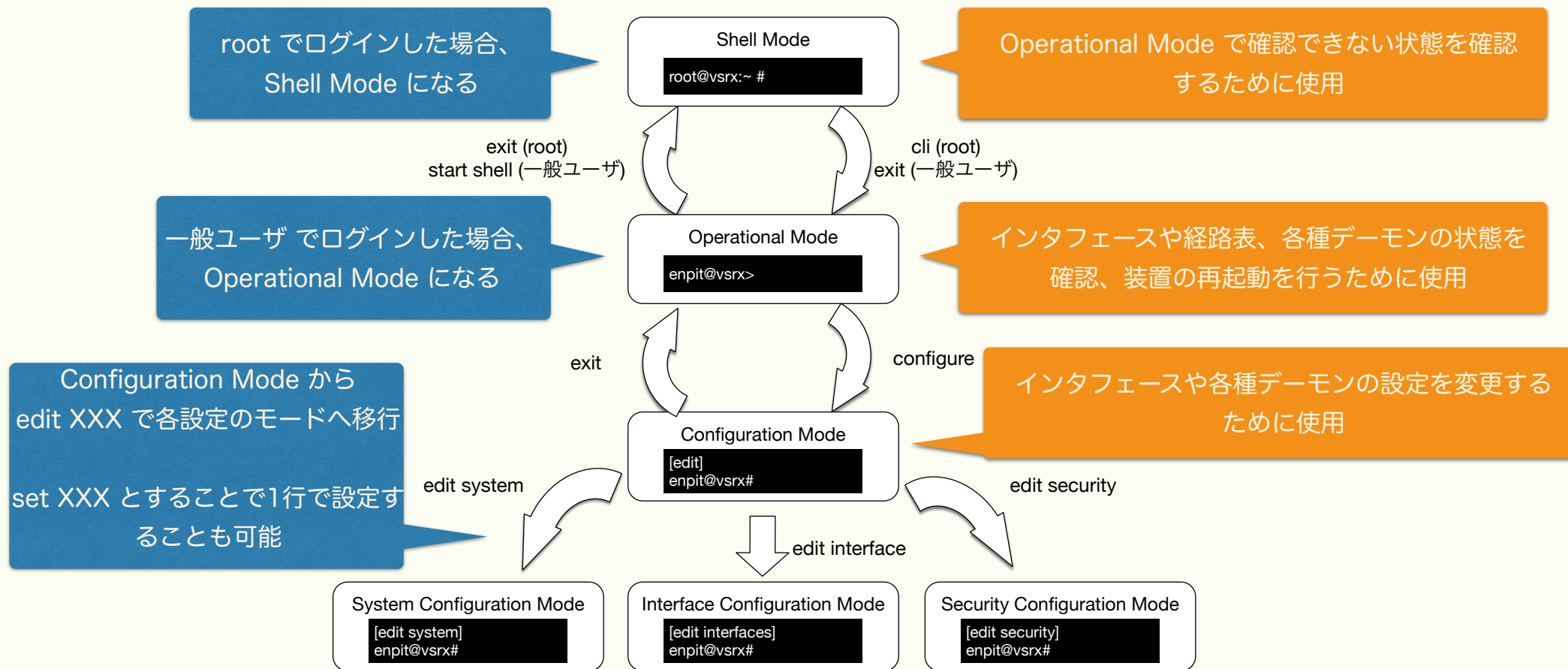
- Juniper が開発している Firewall 製品
 - Firewall 以外にもルータとしても扱える
 - IDS/IPSアンチウイルス
 - アンチスパム
 - Web フィルタリングなどが可能
- OS として Junos が動作
- 今回の演習のセキュリティデバイス



JUNOS

- Juniper 製品で使用する Network OS
 - Juniper 製品はほぼすべて JUNOS で設定
- SRX は用途によってパケット転送方法を選択
 - Flow-based forwarding: ステートフルインスペクションモード
 - Packet-based forwarding: ルータモード
- telnet/ssh、http/https、シリアルコンソール接続で操作が可能
 - 今回は telnet/ssh、シリアルコンソールのみ
- Junos は初期状態ではほぼ何も設定されていない
 - ユーザアカウント、root パスワード、管理ネットワークから設定を行う

JUNOS: Shell / Operational / Config モード



Junos: ログイン

- コンソールでアクセス
 - 初期化後は root でパスワードなし
 - ログイン後は最初に root のパスワードを設定すること
 - パスワードを設定しないと設定を反映できない
- root でログインすると shell が起動
 - Operational Mode へは cli コマンドで移行
 - 設定を投入するためには configure
- 一般ユーザでログインすると cli モード
 - shell に移行する場合は start shell

Junos: 設定フォーマット

- 各項目ごとに階層化

- system
- interfaces
- protocols など

```
enpit@vsrx-1> show configuration | no-more
system {
    /* ホスト名やユーザ、ログ関連などのシステムパラメータの設定 */
}
security {
    /* セキュリティ関連の設定
        ゾーン間のポリシーやフィルタリング、IDS/IPS、IPsec や pki などの設定
        NAT、VPN に関する設定
    */
}
interfaces {
    /* インタフェースの設定 */
}
policy-options {
    /* ルーティングプロトコル、FIB などに対するポリシーの設定 */
}
protocols {
    /* OSPF、BGP などルーティングプロトコルの設定 */
}
routing-options {
    /* rib の操作や flow の制御などパケットフォワーディングに関する設定 */
}
routing-instance {
    /* VRF や仮想ルータ、vpn に関する設定 */
}
```


Junos: 設定の追加方法

- 設定の追加は set で行う
 - 階層の key を入れると次の階層に下がる

- 例:

- ユーザの追加は

set system login user enpit class super-user

設定の追加

```
system {  
    // 省略  
    login {  
        user enpit {  
            uid 2000;  
            class super-user;  
            authentication {  
                encrypted-password "XXX"; ## SECRET-DATA  
            }  
        }  
    }  
    // 省略  
}
```

Junos: 設定の削除方法

- 設定の追加は delete で行う
 - 階層の key を入れると次の階層に下がる

- 例:

- ユーザの削除は

delete system login user enpit

- user enpit 以下がすべて削除される

```
system {  
    // 省略  
    login {  
        user enpit {  
            uid 2000;  
            class super-user;  
            authentication {  
                encrypted-password "XXX"; ## SECRET-DATA  
            }  
        }  
    }  
    // 省略  
}
```

この範囲の設定が削除される

Junos: 設定を一時無効にする方法

- 設定の無効化は deactivate で行う
 - 階層の key を入れると次の階層に下がる
- 例:
 - ユーザの無効化は
deactivate system login user enpit
 - user enpit 以下がすべて無効にされる
- 無効にされている設定は、inactive: で表現される
 - 再度有効にする場合は、
activate system login user enpit

```
system {  
  // 省略
```

```
login {
```

```
  inactive: user enpit {  
    uid 2000;  
    class super-user;  
    authentication {  
      encrypted-password "XXX"; ## SECRET-DATA  
    }  
  }  
}
```

```
  // 省略
```

```
}
```

この範囲の設定が無効になる
(設定は消えない)

Junos: 設定の commit

- 設定の反映
 - Configuration モードで行った設定は Candidate Configuration
 - 設定を反映させるためには commit コマンドを実行
 - commit 実行時に設定が正しいかを検証
 - 事前に検証したい場合は commit check を実行
 - commit confirmed X と実行すると、X 分後に元の設定に戻る
 - 再度 commit を実行すれば永続的に反映
 - 変更後の動作に不安がある場合に使用
- 設定の rollback
 - commit した設定は最大 50 世代まで保存
 - rollback コマンドで過去の設定に切り戻しが可能

```
[edit]
enpit# delete security screen

[edit]
enpit# commit check
[edit security zones security-zone untrust screen]
'screen untrust-screen'
referenced ids-object must be defined under [security screen ids-option]
error: configuration check-out failed: (statements constraint check failed)
```

security の screen を削除した後、commit すると。。

security zones security-zone untrust で screen
の設定を参照しているため commit できない
メッセージが表示される

Junos: 設定の rollback

- 設定のロールバック
 - ある時間に commit したときの戻すコマンド
 - デフォルトでは 50 世代管理される
- rollback X でその時の設定が
Candidate Configuration に反映される
 - 直前の設定は rollback 0

```
[edit]
enpit# rollback ?
Possible completions:
  <[Enter]>      Execute this command
  0              2020-10-12 04:59:52 UTC by enpit via cli
  1              2020-10-12 04:59:45 UTC by enpit via cli
  2              2020-10-12 04:59:21 UTC by enpit via cli
  3              2020-10-12 04:59:17 UTC by enpit via cli
  4              2020-10-12 04:52:53 UTC by enpit via cli
  5              2020-10-12 04:52:48 UTC by enpit via cli
  6              2020-10-12 04:51:51 UTC by enpit via cli
  7              2020-10-12 04:51:46 UTC by enpit via cli
  8              2020-10-12 04:49:32 UTC by enpit via cli
  9              2020-10-09 05:54:24 UTC by root via cli
  10             2020-10-09 03:56:52 UTC by root via cli
  11             2020-10-09 03:33:53 UTC by root via other
  |              Pipe through a command
```

Junos: 設定の rollback compare

- 保存されている過去の設定の差分を表示
 - show system rollback compare X Y
 - X から Y への変更を表示
 - 追加された設定は +、削除された設定は -

```
enpit> show system rollback compare 10 0
[edit system]
+   login {
+       user enpit {
+           uid 2000;
+           class super-user;
+           authentication {
+               encrypted-password "$6$RE0
31BzdyWM2QpaM/"; ## SECRET-DATA
+           }
+       }
+   }
[edit system services ssh]
+   root-login deny;
[edit system services]
+   telnet;
```

JUNOS: 設定に困ったときは？

- ?を入力すると、そこから入力できるコマンドの一覧が表示される

```
[edit]
enpit# show system login ?
Possible completions:
  <[Enter]>      Execute this command
  announcement   System announcement message (displayed after login)
+ apply-groups   Groups from which to inherit configuration data
+ apply-groups-except Don't inherit configuration data from these groups
> class          Login class
> deny-sources   Sources from which logins are denied
  idle-timeout    Maximum idle time before logout (1..60 minutes)
  message         System login message
> password       Password configuration
> retry-options  Configure password retry options
> user           Username
  |              Pipe through a command
```

これらのコマンドが

次に入力可能なコマンド

JUNOS: ユーザの追加

- 初期設定では root 以外のユーザは存在しない
 - root もパスワードは未設定
 - root のパスワードを設定しないと commit 時にエラー
- ユーザ作成にはクラスとパスワードを設定
 - クラスはユーザの権限
 - super-user: すべての権限
 - operator: パーミッションのクリア、ネットワーク、リセット、トレース
 - read-only: パーミッションの参照
 - unauthorized: パーミッションなし
 - super-user は常に1つ以上必要
 - パスワードは最低 6 文字、大文字を含める

JUNOS: ユーザ追加の例

- configure mode で作業
 - ユーザ名を指定して class とパスワードを入力
 - super-user は特権ユーザ
 - パスワードは plain-text-password で入力
 - 入力後、暗号化される

```
enpit@vsrx> configure
Entering configuration mode

[edit]
enpit@vsrx# set system login user enpit class super-user

[edit]
enpit@vsrx# set system login user enpit authentication plain-text-password
New password:
Retype new password:

[edit]
enpit@vsrx# show system login
user enpit {
    uid 2000;
    class super-user;
    authentication {
        encrypted-password "XXX"; ## SECRET-DATA
    }
}

[edit]
enpit@vsrx# commit
```

JUNOS: その他基本的な設定

- ホスト名
 - set system host-name で設定
- telnet/ssh の設定
 - set system service telnet
 - set system services ssh root-login deny
 - root でのログインを禁止
- ntp の設定
 - set system ntp server X.X.X.X
 - X.X.X.X のサーバに同期のリクエストを送信する
 - set system ntp source-address
 - ntp での時刻同期に使用するアドレスの指定

JUNOS: インタフェース名

- XX-0/0/0 などのような命名
 - LinkSpeed-Chassis 番号/Slot 番号/Port 番号
- Link Speed によって名前が異なる
 - 100Mbps: fe-0/0/0 (Fast Ethernet)
 - 1Gbps: ge-0/0/0 (Gigabit Ethernet)
 - 10Gbps: xe-0/0/0 (10 Gigabit Ethernet)
 - 40/100Gbps: et-0/0/0 (まとめて et へ変更になった)
- 管理用インタフェースは fxp0
 - 製品によっては me0、em0 もある
- トンネル用に gr (GRE)、ip (IP-over-IP)、pp (PPPoE)
- 冗長化用に fab (Control Plane)、reth (Redundant Ethernet)、ae (Aggregate Ethernet) などがある

JUNOS: インタフェース設定

- set interface INTERFACE unit X family FAMILY で設定
 - INTERFACE は ge-0/0/0 などのインタフェース名
 - unit X で論理インタフェースを管理
 - unit は VLAN を使用しない場合、0 を指定
 - VLAN を使用する場合は 0 以外を指定
 - VLAN ID と合わせておくとうわかりやすい
 - family はプロトコルファミリを指定
 - IPv4 であれば inet
 - IPv6 であれば inet6
 - その他、ethernet-switching、mpls などがあるが今回は使用しない
- inet、inet6 の場合、以降 address で IP アドレスを指定
- ディスクリプションの設定
 - set interface description
 - set interface unit X description

JUNOS: インタフェース設定の例

- ge-0/0/0 を設定
 - 何の用途に使用しているかわかりやすくするため description を設定
 - IPv4 アドレス(family inet)、192.168.10.1/24 を設定

```
[edit]
enpit@vsrx# set interfaces ge-0/0/0 description external

[edit]
enpit@vsrx# set interfaces ge-0/0/0 unit 0 family inet address 192.168.0.1/24

[edit]
enpit@vsrx# show interfaces ge-0/0/0
ge-0/0/0 {
    description external;
    unit 0 {
        family inet {
            address 192.168.10.1/24;
        }
    }
}

[edit]
enpit@vsrx# commit
```

Junos: Firewall の設定

- 2種類の設定方法
 - set firewall
 - ACL、主にルータを守るために使用
 - その他、Policy-based Routing など
 - set security
 - Firewall としてフィルタリングする場合に使用
 - zone で管理可能

Junos: Firewall の設定

- edit firewall family inet filter FILTER
 - set term telnet from source-address IPADDRESS/PREFIX
 - set term telnet from destination-port telnet
 - set term telnet then accept
-
- set interface lo0 unit 0 family inet filter input FILTER
 - commit

Junos: セキュリティゾーン

- インタフェースに割り当てる仮想的なグループ
 - SRX では主にセキュリティゾーンでトラフィックを制御
- Functional zone
 - 管理系のためのゾーン
- Security zone
 - FW を通過するトラフィックを制御するゾーン
- デフォルトで trust、untrust ゾーンが設定されている

```
enpit@vsrx> show configuration security zones
security-zone client {
    host-inbound-traffic {
        system-services {
            ping;
            traceroute;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
```

zone の設定例

Junos: セキュリティゾーン設定

- security zone には interface、host-inbound-traffic の設定を行う
- interface: security zone で管理するインタフェースを指定
 - ge-0/0/0.0 のように指定する
 - 最後の .0 は unit 番号。この例では unit 0
- host-inbound-traffic: SRX 宛のトラフィック
 - system-services
 - zone で通過させるアプリケーションを指定
 - protocols
 - zone で通過させるプロトコルを指定

Junos: ポリシー

- zone 間のトラフィックを制御
 - from-zone から to-zone へ片方向の設定を記述
 - match 内が条件
 - source-address
 - destination-address
 - source, destination は address book 名で指定
 - application
 - application はデフォルトで定義されているものがある
 - show group junos-defaults で確認可能
 - then がアクション
 - permit と deny、log でセッションのログを記録

```
security {  
  policy {  
    from-zone external to-zone client {  
      policy icmp {  
        match {  
          source-address any;  
          destination-address any;  
          application junos-icmp-all;  
        }  
        then {  
          permit;  
        }  
      }  
      policy default-deny {  
        match {  
          source-address any;  
          destination-address any;  
          application any;  
        }  
        then {  
          deny;  
          log {  
            session-init;  
            session-close;  
          }  
        }  
      }  
    }  
  }  
}
```

policy の設定例

Junos: Address book

- 各ゾーンで管理するアドレスをまとめたもの
 - wiki: 192.168.100.101 など
- set security zone security-zone 以下に設定
 - policy の source-address、destination-address に使用

```
zones {  
    security-zone dmz {  
        address-book {  
            address wordpress 192.168.100.101/32;  
            address wiki 192.168.100.102/32;  
        }  
    }  
}
```

各サーバのアドレス

- external: 192.168.0.0/24
 - SRX: 192.168.0.1
 - wordpress: 192.168.0.101
 - attacker: 192.168.0.102
- DMZ: 192.168.100.0/24
 - SRX: 192.168.100.1
 - wordpress: 192.168.100.101
 - wiki: 192.168.100.102
 - ftp: 192.168.100.103
 - samba: 192.168.100.104
 - ntp: 192.168.100.105

各サーバの mgmt アドレス

- SRX: 10.10.0.1 (設定されていないので、このアドレスを設定すること)
- wordpress: 10.10.100.101
- wiki: 10.10.100.102
- ftp: 10.10.100.103
- samba: 10.10.100.104
- ntp: 10.10.100.105
- log: 10.10.100.106

ユーザ名とパスワード

- 各種サーバ
 - enpit/Enpitpro

演習1: vSRX の基本設定 (1/2)

- vSRX-NG の基本設定を行う
 - root ユーザのパスワード設定
 - enpit ユーザの作成
 - パスワードはともに Enpitpro
 - ホスト名を vsrx に設定
 - 管理インタフェース (fxp0) の設定
 - fxp0 のアドレスは 10.10.0.1/16
 - telnet、ssh の設定
 - ssh は root でのログインを禁止すること
- これらを行った後、enpit ユーザで telnet、ssh できることを確認する

演習1: vSRX の基本設定 (2/2)

- vSRX-NG のネットワーク設定を行う
 - 各インタフェースの description、IP アドレス、security zoneの設定を行う
 - security zone は以下のパケットを許可
 - system-services
 - ping
 - traceroute
 - protocols
 - すべて許可しない
 - これらの設定をした上で各ネットワークのサーバ/クライアントから ping での疎通確認を行う
- ge-0/0/0
 - description: external
 - ipv4 address: 192.168.0.1/24
 - security zone: external
- ge-0/0/1
 - description: dmz
 - ipv4 address: 192.168.100.1/24
 - security zone: dmz
- ge-0/0/2
 - description: client
 - ipv4 address: 192.168.200.1
 - security zone: client

演習2: Firewall/Router の設定 (1/4)

- 演習のネットワークで以降説明する条件で通信可能なよう設定を行う
 - 各 zone 間の通信に対してポリシーを設定する
 - 必要があれば、address-book も設定すること
- 設定後、各ネットワークのサーバ/クライアントから ping やブラウザを用いて通信可能か、通信不可能のとなっているかを確認する
- ただし、同じ zone 間の確認は行わなくてよい
 - external -> external など

演習2: Firewall/Router の設定 (2/4)

- External からのポリシーを以下のように設定する
 - external -> external
 - 許可しない
 - external -> dmz
 - ping、traceroute を許可
 - wordpress には tcp port 80 を許可し、session-init、session-close のログを取る
 - external -> client
 - icmp のみ許可

演習2: Firewall/Router の設定 (1/4)

- DMZ からのポリシーを以下のように設定する
 - dmz -> external
 - ping、traceroute のみ許可
 - dmz -> dmz
 - 許可しない
 - dmz -> client
 - icmp のみ許可

演習2: Firewall/Router の設定 (1/4)

- Client からのポリシーを以下のように設定する
 - client -> external
 - すべて許可、ただし session-init、session-close のログを取ることに
 - client -> dmz
 - 以下のサーバ、ポートへのアクセスを許可
 - wordpress: http
 - wiki: http
 - ftp: ftp
 - ntp: ntp
 - client -> client
 - すべて許可

演習3-1: WAF の設定

- wordpress にて WAF を動作させ、不正なURL を検知することを確認する
 - apache mod_security はインストール済み
 - /etc/httpd/conf.d/mod_security.conf で動作を検知のみに変更して wordpress にアクセスできることを確認する
 - また、/var/log/httpd/error_log に検知ログを確認する
- どのような理由で mod_security が通信をブロックしているかを調べる

演習3-2: wordpress の脆弱性診断

- client の Kali Linux にインストールされている wpscan を用いて wordpress の脆弱性診断を行う
 - 脆弱性診断のコマンドは以下の通り
 - `wpscan --url 192.168.100.101`
 - wpscan にて検出された脆弱性に関して
 - 原因はなにか
 - 対処はどのようにすべきかを考察する

演習4: Syslog を用いたログ監視

- log サーバで rsyslog を用いて、vsrx、wordpress のログを収集する
 - vsrx は、演習2で取得している session-init、session-close のログを転送
 - wordpress は apache のログを転送
- log サーバでは、vsrx、wordpress からのログを、以下のファイルに保存
 - vsrx: /var/log/enpit/vsrx.log
 - wordpress: /var/log/enpit/wordpress.log

演習4: Syslog を用いたログ監視

- rsyslog によるログの待受
 - rsyslog の設定ファイルは /etc/rsyslog.conf
 - 各サーバの設定は /etc/rsyslog.d/ に ~~.conf で作成
- rsyslog.conf
 - syslog は UDP: 514 で転送されるため、これを listen するよう設定を行う

```
module(load="imuxsock") # provides support for local system logging
#module(load="immark") # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514") ←コメントアウトを削除
```


演習5: SNMP を用いたトラフィック監視

- log サーバにインストールされている cacti を用いて vsrx のトラフィックを監視する
 - monitor で firefox を起動し、以下の url にアクセス
 - `http://10.10.0.106/cacti`
 - ユーザ名: admin、パスワード: Enpitpro
- snmp コミュニティを enpit とし、vsrx の各インタフェースのトラフィックを監視

演習5: SNMP を用いたトラフィック監視

- vsrx の設定
 - snmp 以下にコミュニティ、アクセス制限、権限を設定

- 設定例
 - clients のアドレスは各環境で変更すること
 - 他のノードからアクセスできないようにすること
 - 他のノードから書き換えできないようにすること

```
[edit]
root# show snmp
interface fxp0.0;
community enpit {
    authorization read-only;
    clients {
        100.64.209.106/32;
        0.0.0.0/0 restrict;
    }
}
```

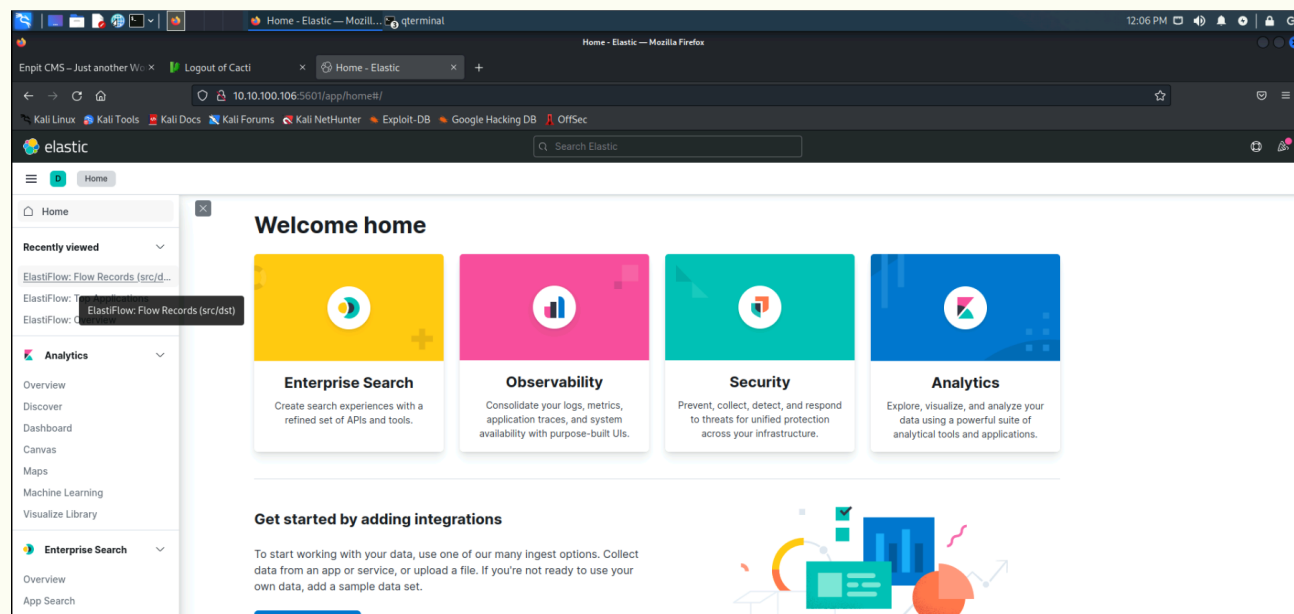
演習6: Netflow によるトラフィック監視

vSRX の設定

- `set forwarding-options sampling family inet input rate 1024`
 - Sampling rate を 1024 に設定
- `set services flow-monitoring version9 template ipv4 ipv4-template`
 - Netflow version9 のテンプレートを作成
- `set services flow-monitoring version9 template ipv4 flow-key flow-direction vlan-id`
 - 作成したテンプレートで取得する flow-key を設定
- `set forwarding-options sampling family inet output flow-server 10.10.100.106 port 2055`
 - NetFlow の送信先を設定
- `set forwarding-options sampling family inet output flow-server 10.10.100.106 version9 template ipv4`
 - NetFlow の送信先に適用するテンプレートを指定
- `set forwarding-options sampling family inet output inline-jflow source-address 10.10.0.1`
 - NetFlow データを送信する際の送信元アドレスを指定

演習6: Netflow によるトラフィック監視

- <http://10.10.100.106:5601> にアクセス
- ElasticFlow: FlowRecords src/dst をクリック



その他

- 演習中にわからない点、うまくいかない点があれば質問すること
- レポートの内容
 - 各演習課題で行った設定、確認したことをスクリーンショットを用いて説明すること
 - 特別に考慮した点があればそれも説明してもよい
- 演習環境は 11/30 まで利用可能