

LPICレベル2技術解説セミナー

2012/1/16

株式会社ジェイ・ジェイ・エス
研修教育事業部
大森 聡



➤ 会社概要

株式会社ジェイ・ジェイ・エス(<http://www.jjs.co.jp/>)

➤ 講師紹介

研修教育事業部所属 大森 聡

Linuxおよびネットワーク・セキュリティ系をメインに、ベンダー系資格対策講習、情報処理技術者試験講座、新人SE教育研修など、様々な技術研修を担当



本日のアジェンダ



➤ LPIレベル2 受験の心構え

LPI認定資格とは
試験の範囲
学習環境の構築
受験のポイント

➤ 技術的な詳細解説

LPI201試験範囲より頻出ポイントを解説

休憩(10分程度)

LPI202試験範囲より頻出ポイントを解説

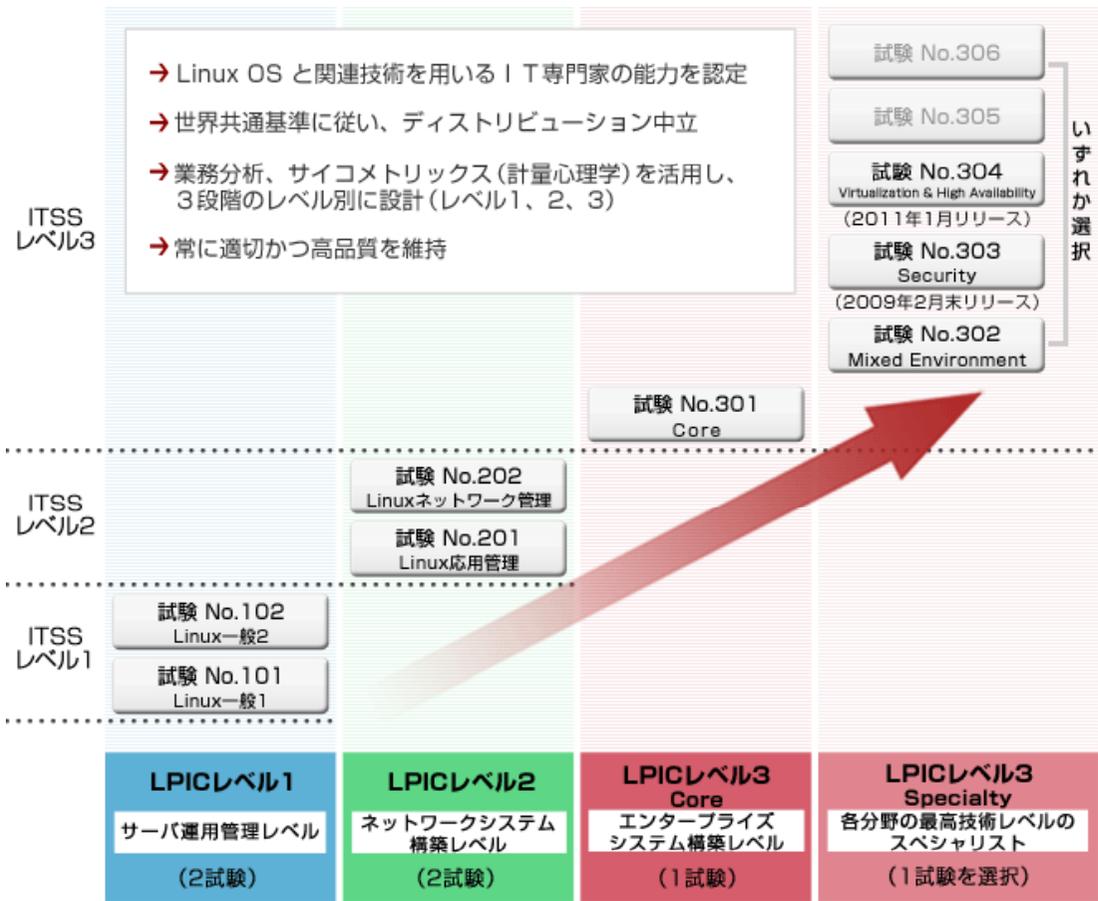


- ベンダーニュートラルなグローバル認定資格
NPOであるLinux Professional Instituteが実施
- Linux技術者のスキルを計る一つの目安
他にRHEC, Turbo-CE, CompTIA(Linux+)などの資格がある
- レベル1からレベル3までの3段階に分れる
レベル1認定 - LPI Level1 Exam101,102に合格
レベル2認定 - レベル1認定に加え、LPI Level2 Exam201,202に合格
レベル3Core認定 - レベル2認定に加え、LPI 301 Core Examに合格

※その他、LPIC302, 303, 304の各Speciality認定があります



LPI認定の構成





201試験の概要



- 出題数: 約60~75問
- 制限時間: 90分
- 合格ライン: 65%程度
- 試験範囲
 - 主題201: Linuxカーネル
 - 主題202: システムの起動
 - 主題203: ファイルシステムとデバイス
 - 主題204: 高度なストレージ管理
 - 主題205: ネットワーク構成
 - 主題206: システムの保守
 - 主題207: ドメインネームサーバ(DNS)



202試験の概要



- 出題数: 約60~75問
- 制限時間: 90分
- 合格ライン: 65%程度
- 試験範囲
 - 主題208: Webサービス
 - 主題209: ファイルとサービスの共有
 - 主題210: ネットワーククライアントの管理
 - 主題211: 電子メールサービス
 - 主題212: システムのセキュリティ
 - 主題213: トラブルシューティング



➤ 学習教材

定番の市販のテキスト、問題集、インターネット上の資料などを揃える

設定ファイル名、コマンド名、コマンドオプション、コマンドの実行結果、ログ等については、実機でも確認する

➤ 実機環境

CentOS, **Feroda**, **Debian**, **Ubuntu**などのディストリビューションがおすすめ

メモリに余裕のあるマシン上に、**VM Player**, **Virtual Box**, **Virtual PC**などの無償の仮想化ツールを導入する

仮想化ツールを利用して、同時に複数台のLinuxを稼働させると、クライアント-サーバ系のアプリケーションの動作検証ができる



➤ 試験形式

CBT形式で、択一問題、複数選択問題、入力問題の3パターンがある

※入力形式の問題は十数問出題される ←合否のポイント

➤ 試験テクニック

出題文の意図を正確に読み取る

できる問題を確実に、わからない問題はマークして後回しにする

入力形式の問題はスペルミスに気をつける

選択肢で迷った場合は、消去法を使う

最後に、すべての解答を見直す



➤ LPICレベル2試験のポイントとなる部分について重点的に解説します

試験で出題されやすいコマンド、設定ファイル等

覚えておくべき必須の知識項目

勉強する際に、混乱しやすい技術的な事柄

試験には直接出題されないが、Linux技術者として押さえておきたいテクニック



➤カーネルモジュールとは？

カーネルを構成する機能を、**分離可能な部品**として独立させたもの。ロードブルモジュールとも呼ぶ。おもにデバイスドライバがカーネルモジュールとして使用される。

➤カーネルモジュールが配置されるディレクトリ

/lib/modules/カーネルバージョン/

➤**modprobe**コマンド

モジュールの依存関係を調べて、モジュールのロード/アンロードを実行する

➤**modules.dep**ファイル

モジュール間の依存関係が記述されたファイル。**depmod**コマンドで生成する

※学習ポイント:各コマンド、設定ファイルの役割、関連をしっかりと把握する



➤カーネルのコンパイルとインストール手順

1. カーネルソースを用意する(/usr/src/linux/)
2. カーネルコンフィグレーションを設定する(make configコマンド)
3. カーネル本体とカーネルモジュールをコンパイルする(makeコマンド)
4. カーネルモジュールをインストールする(make modules_install)
5. カーネル本体をインストールする(make install)
6. ブートローダ(GRUBまたはLILO)の設定を変更する

➤カーネルコンフィグレーションファイル

.configファイル .. モジュール毎に、カーネルに静的に組み込む(Y)か、動的に組み込む(M)か、あるいは組み込まない(N)かを指定する

make config, make menuconfig, make xconfig等のコマンドで編集が可能

※学習ポイント:カーネル構築の各段階でどのコマンドが必要なのか整理する



■Linuxの起動シーケンス

1. BIOS(EFI)の実行
2. ブートローダ(GRUBまたはLILO)の実行
3. カーネルのロード
4. initプロセスの実行 → **/etc/inittab**ファイルを参照
5. 各種rcスクリプトの実行
6. ランレベルに対応したサービスの実行

■サービスの手動制御 ※sshの場合

/etc/init.d/sshd start | stop | restart | reload | status | ..etc

■サービスの自動起動

chkconfigまたは**ntsysv**(Redhat系), **update-rc.d**(Debian系),**insserv**(Suse系)などのランレベルエディタを利用する

※学習ポイント:LPIレベル2では、**/etc/inittab**の詳細知識が問われる



■ ファイルシステムの作成手順

1. パーティションの作成 → `fdisk` コマンド
2. ファイルシステムの作成 → `mke2fs` コマンド
3. マウント/アンマウント → `mount / umount` コマンド

■ ファイルシステムの保守

- ファイルシステムのチェック → `fsck, e2fsck` コマンド
ファイルシステムのパラメータ変更 → `tune2fs` コマンド
ファイルシステムの詳細確認 → `dumpe2fs` コマンド

■ スワップ領域の作成手順

1. スワップ領域の作成 → `mkswap`
2. スワップ領域の有効/無効化 → `swapon / swapoff`

※学習ポイント: 各コマンドのオプションについても要注意



■ 各コマンドのオプション(抜粋)

`mke2fs -j` → ext3ファイルシステムを作成する

`mke2fs -m` → rootユーザ用の予約領域を指定する

`e2fsck -p` → 不良ブロックを自動的に修復する

`tune2fs -i` → ファイルシステムのチェック間隔を変更する

`tune2fs -j` → ext2からext3へ変換する

`tune2fs -m` → rootユーザ用の予約領域を変更する

`mount -o remount` → 再マウントする

`mount -o loop` → ループバックマウントをする

`mkisofs -J` → Jolietフォーマット(Windows系OSで採用)でISO9660ファイルイメージを生成する

※学習ポイント: コマンドオプションについては、可能な限り実機で試してみる



■ LVM(論理ボリューム管理)の特徴

- 一度作成したパーティションのサイズ変更が可能
- パーティションを別ディスクに移動することが可能
- 複数ディスクにまたがるパーティション作成が可能

■ LVMの作成手順

1. 従来のパーティションを物理ボリュームとして設定する(**pvcreate**コマンド)
2. 物理ボリュームからボリュームグループを作成する(**vgcreate**コマンド)
3. ボリュームグループ内から論理ボリュームを作成する(**lvcreate**コマンド)

作成した論理ボリュームは、従来のパーティションと同様に、**mke2fs**コマンドでファイルシステムを作成し、**mount**コマンドで特定ディレクトリにマウントすることができる

※学習ポイント:LVM関連のコマンド群を整理しておく。LVM管理コマンドとして、**vgextend** / **vgreduce**コマンド(ボリュームグループの拡張/縮小)も要注意



■ ネットワーク関連コマンド

ifconfig .. ネットワークインタフェースの設定、表示

arp .. ARPエントリの表示(-a)、設定(-s)、削除(-d)

ping .. ICMPを使ったネットワークの疎通確認

traceroute .. ICMPを使ったネットワークの経路表示

tcpdump .. ネットワークを通過するパケットの捕捉

netstat .. 各種ネットワーク情報の表示: 接続待ちソケット(-l)、ルーティングテーブル(-r)、パケット統計情報(-s)

route .. ルーティングテーブルの設定、表示

※学習のポイント: 各コマンドの実行結果、オプション等についても要確認



■ ネットワーク関連ファイル

/etc/hosts .. IPアドレスとホスト名の対応関係

(例) 192.168.0.1 server.example.com server

/etc/networks .. ネットワーク名とネットワークアドレスの対応関係

(例) localnet 192.168.0.0

/etc/nsswitch.conf .. 名前解決を行う際の問い合わせの順序

(例) hosts: files ldap dns

/etc/resolv.conf .. 問い合わせ先のDNSサーバとドメイン名

(例) search example.com
nameserver 192.168.0.2

※学習のポイント: 各種ネットワーク設定ファイルとその書式を押さえる



■ ソースアーカイブの展開

※software.tar.gzファイルを解凍、展開するコマンド(3パターン)

- a. **tar zxvf** software.tar.gz
- b. **gunzip** software.tar.gz ; **tar xvf** software.tar
- c. **gzip -dc** software.tar.gz | **tar xvf -**

※software.tar.bz2ファイルを解凍、展開するコマンド(3パターン)

- d. **tar jxvf** software.tar.bz2
- e. **bunzip** software.tar.bz2 ; **tar xvf** software.tar
- f. **bzip -dc** software.tar.bz2 | **tar xvf -**

※学習のポイント: tarコマンドのオプション、とくに-c,-x,-t,-f,-z,-jについて機能を押さえておく



■ ソースアーカイブからのインストール手順

※software.tar.gzをインストールする場合

1. `tar zxvf software.tar.gz` →解凍&展開
2. ドキュメント(README,INSTALL等)の参照
3. `./configure` →Makefileの生成
4. `make` →コンパイル
5. `su` →root(管理者)権限を取得
6. `make install` →インストール

※学習ポイント:./configureのスペル、インストール時に管理者権限が必要になる点に注意



■ リソースレコードと名前解決

Aレコード:ホスト名→IPアドレス (正引き)

PTRレコード:IPアドレス→ホスト名 (逆引き)

MXレコード:ドメイン名→メールサーバのホスト名

NSレコード:ドメイン名→DNSサーバのホスト名

CNAMEレコード:別名→正規のホスト名

リソースレコードおよび名前解決に関する情報が記述されたファイルを”ゾーンファイル”と呼ぶ

■ DNSサーバの構成

マスターサーバ:ゾーンファイルのオリジナルを持つ

スレーブサーバ:ゾーンファイルのコピーを持つ

マスターサーバからスレーブサーバへゾーンファイルが転送されることを”**ゾーン転送**”と呼ぶ



■DNSの実装

パッケージ名: BIND

デーモン名: named

設定ファイル: /etc/named.conf

■named.confの設定例

```
options { directory "/var/named"; };
```

```
zone "." { type hint; file "named.ca"; }; //ルートヒントファイル
```

```
zone "localhost" { type master; file "localhost.zone"; }; //ローカルホストの正引き
```

```
zone "0.0.127.in-addr.arpa" { type master; file "localhost.rev"; }; //ローカルホストの逆引き
```

```
zone "example.com" { type master; file "example.zone"; }; //example.comゾーンの正引き
```

```
zone "0.168.192.in-addr.arpa" { type master; file "example.rev"; }; //example.comゾーンの逆引き
```



■ 正引きゾーンファイルの例

\$TTL 86400

```
@      IN      SOA      ns1.example.com.  root.example.com. (
        2012202201 ; Serial
        10800      ; Refresh
        600        ; Retry
        3600000    ; Expire
        86400 )    ; Negative TTL

        IN      NS       ns1.example.com.
        IN      NS       ns2.example.com.
        IN      MX 10    mail1.example.com.
        IN      MX 20    mail2.example.com.

ns1     IN      A        192.168.0.1
ns2     IN      A        192.168.0.2
mail1   IN      A        192.168.0.3
mail2   IN      A        192.168.0.4
host    IN      A        192.168.0.5
www     IN      CNAME    host.example.com.
```



■ 逆引きゾーンファイルの例

\$TTL 86400

```
@      IN      SOA      ns1.example.com.  root.example.com. (
                                2012202201 ; Serial
                                10800      ; Refresh
                                600        ; Retry
                                3600000   ; Expire
                                86400 )   ; Negative TTL
```

```
      IN      NS      ns1.example.com.
      IN      NS      ns2.example.com.
```

```
1     IN      PTR     ns1.example.com.
2     IN      PTR     ns2.example.com.
3     IN      PTR     mail1.example.com.
4     IN      PTR     mail2.example.com.
5     IN      PTR     host.example.com.
```

※学習のポイント:ゾーンファイルの書式については詳細を理解しておく



■ Webサーバの実装

パッケージ名: Apache

デーモン名: httpd

設定ファイル: /etc/httpd/conf/httpd.conf

■ httpd.confの設定例(抜粋)

| | | |
|-----------------------|----------------------|----------------------|
| DocumentRoot | /var/www/html | #ドキュメントルート(公開ディレクトリ) |
| UserDir | public_html | #一般ユーザの公開ディレクトリ |
| DirectoryIndex | index.html index.htm | #インデックスファイル |
| ErrorLog | logs/error_log | #エラーログファイル |
| Alias | /img /var/www/img | #エイリアス(公開ディレクトリの別名) |
| AccessFileName | .htaccess | #外部設定ファイル |
| AllowOverride | AuthConfig Limit | #外部設定ファイルによる許可項目 |

※学習ポイント: httpd.confのディレクティブ(設定項目)は頻出のため、確実に押さえておく



■プロキシサーバの実装

パッケージ名: Squid

デーモン名: squid

設定ファイル: /etc/squid/squid.conf

役割: コンテンツキャッシュ、アクセス制御

■squid.confの設定例(抜粋)

```
http_port 8080 #squidが利用するポート番号
cache_dir ufs /var/spool/squid 100 16 256 #キャッシュを保存するディレクトリと容量
reply_body_max_size 0 #レスポンスの最大ボディサイズ

acl localnet src 192.168.0.0/255.255.255.0 #アクセスリストの定義
acl blacklists url_regex "/etc/squid/url_blacklist.txt"

http_access allow localnet #アクセスの許可/拒否
http_access deny blacklists
```

※学習ポイント: squid.confのディレクティブ(設定項目)について押さえておく



■ファイルサーバの実装①

パッケージ名: Samba

デーモン名: smbd(ファイル共有、認証), nmbd(ブラウジング、名前解決)

設定ファイル: /etc/samba/smb.conf

役割: Linuxホストを、Windowsのファイル/プリントサーバとして代用する

■smb.confの設定例(抜粋)

| | |
|--|-------------------------------|
| unix password sync = Yes No | #WindowsとLinuxのパスワードを同期させるか否か |
| username map = マッピングファイル名 | #WindowsとLinuxのユーザアカウントを対応づける |
| browseable = Yes No | #コンピュータブラウジングした際に、表示するか否か |
| ※共有名の前に"\$"をつけて、ブラウザリストで非表示にすることも可能(管理共有) | |
| writable = Yes Noまたは read only = No Yes | #ファイルの書き込みを可能にするか否か |
| hide files = /ファイル名/ | #表示させたくないファイルを指定 |
| veto files = /ファイル名/ | #表示もアクセスもさせたくないファイルを指定 |
| os level = 数値 | #マスターブラウザとして選択される場合のパラメータ |
| security = SHARE USER SERVER DOMAIN AD | #認証方式の指定 |

※学習ポイント: Windowsネットワークの特徴(認証、名前解決、ブラウジング等)について押さえておく



■ファイルサーバの実装②

パッケージ名:NFS

デーモン名:portmap, nfsd, **mountd**

設定ファイル:**/etc/exports**

■NFSサーバ側(exportsファイルの設定例)

/share 192.168.0.0/255.255.255.0(rw) **#/shareディレクトリを192.168.0.0/24ネットワークに対して、読み書き可能で公開**

/pub *example.com(ro) server(rw, no_root_squash) **#/pubディレクトリをexample.comドメインに対しては読み取り専用、ホストserverに対しては読み書き可能で、root権限でのアクセスを許可**

※exportsファイルを記述した後、**exportfs**コマンドを実行して、設定を有効化する

■NFSクライアント側(mountコマンドの実行例)

mount -t nfs server:/pub /mnt/nfs **#server上で公開された/pubを/mnt/nfsにマウントする**

※学習ポイント:NFSサーバ側とNFSクライアント側で行う作業を整理して覚える



■ DHCPサーバの実装

パッケージ名: ISC DHCP

デーモン名: dhcpd

設定ファイル: /etc/dhcpd.conf

■ dhcpd.confの設定例

```
subnet 192.168.0.0 netmask 255.255.255.0 {  
    option routers          192.168.0.1;          #デフォルトゲートウェイアドレス  
    option subnet-mask      255.255.255.0;        #サブネットマスク  
    option domain-name      "example.com";        #ドメイン名  
    option domain-name-servers 192.168.0.1, 192.168.0.2; #DNSサーバアドレス  
    range (dynamic-bootp)   192.168.0.10 192.168.0.100 #アドレスプールの範囲  
    host notepc {  
        hardware ethernet aa:bb:cc:dd:ee:ff;  
        fixed-address 192.168.0.200;  
    }  
}
```

※学習ポイント: dhcpd.confの書式について正確に押さえておく



■ PAM(Pluggable Authentication Modules)

役割: Linuxの各種サービスに対し、一元的な認証機能を提供

設定ファイル: `/etc/pam.d/`内に、サービスごとに作成

■ 設定ファイルの書式

| モジュールタイプ | コントロール | モジュールのパス | 引数 |
|-------------------|-----------------------|-----------------------------|----|
| <code>auth</code> | <code>required</code> | <code>pam_nologin.so</code> | |

※`/etc/nologin`ファイルが存在した場合は、一般ユーザによるログインを拒否する設定

■ PAMの出題ポイント

モジュールタイプの種類と特徴

- `auth` .. ユーザ認証自体を実行
- `account` .. パスワードの有効性、期限チェックなど
- `session` .. ユーザ認証の前後に実行する処理(ログイン記録など)

コントロールの種類と特徴

- `requisite` .. モジュールの実行に失敗したら、すぐ認証を拒否
- `required` .. モジュールの実行に失敗しても、同じタイプのモジュールをすべて実行した後で認証を拒否
- `sufficient` .. モジュールの実行に成功した場合、すぐ認証を許可



■メールシステムの概要

MTA(Mail Transfer Agent) .. SMTPによるメール転送処理を行う

MDA(Mail Delivery Agent) .. ローカルホストでメール配信を行う

MUA(Mail User Agent) .. メールクライアントソフト

POP/IMAPサーバ .. メールクライアントが接続してメールを受信する

■MTAの実装

パッケージ名: Sendmail

設定ファイル: **/etc/mail/sendmail.cf** → メール転送処理の設定

/etc/mail/access → アクセス制御の設定

/etc/aliases → エイリアス(メールアドレスの別名)の設定

パッケージ名: Postfix

設定ファイル: **/etc/postfix/main.cf** → メール転送処理の設定

/etc/postfix/master.cf → デーモンプロセスの動作設定

※学習ポイント: 各MTA実装の設定ファイル名を押さえておく



■ /etc/mail/accessの設定例

| | | |
|-----------|-----------------------------|------------------------|
| 192.168.1 | RELAY | #メールの中継を許可する |
| 192.168.2 | REJECT | #メールの受信を拒否する |
| 192.168.3 | DISCARD | #メールを破棄(=無視)する |
| 192.168.5 | 550 “No Spam Wanted” | #ステータスコードとメッセージ付きで拒否する |

■ /etc/aliasesの設定例

| | | |
|-----------|---------------------------|-----------------------------------|
| testuser: | user | #testuser宛でのメールをuserのメールボックスに配信する |
| testuser: | /filename | #指定したファイルにメール内容を追記する |
| testuser: | command | #指定したコマンドで処理する |
| testuser: | user@domain | #指定したメールアドレスへメールを転送する |
| testuser: | :include:/filename | #指定したファイルに記述された設定を読み込む |

※/etc/aliasesファイルを編集した後は、**newaliases**コマンドで設定を反映させる必要がある

学習のポイント: 各設定ファイルの書式を正確に押さえる



■ Procmailについて

MDA実装の一つ。メールをローカル配信する際、一定のルール(レシピ)に基づいてフィルタリングする機能がある

設定ファイル:各ユーザのホームディレクトリ内の**.procmailrc**

■ .procmailrcの書式

:0 [フラグ] [:ロックファイル]

* 条件式
アクション

条件式で利用できる記号

- ! .. 条件の否定
- < .. 指定したバイト以下の場合、処理する
- > .. 指定したバイト以上の場合、処理する

(次ページに続く)



アクション一覧

/dev/null .. 破棄する
ファイル名 .. 指定したファイルに追記
ディレクトリ名 .. 指定したディレクトリ内に格納
| **command** .. 指定したコマンドで処理する
! メールアドレス .. 指定したメールアドレスに転送する

■.procmailrcの設定例

```
:0  
* ^Subject:.*SPAM.*  
/dev/null
```

→メールのタイトルに”SPAM”の文字が含まれていた場合、メールを破棄する

```
:0  
* < 5000  
!user@example.com
```

→メールのサイズが5000バイト以内の場合、user@example.comに転送する

※学習のポイント:Procmailのレシピは頻出項目



■ パケットフィルタリング

IPパケットヘッダの送信元/先IPアドレス・送信元/先ポート番号などに基づいてパケットの通過を許可/拒否するしくみ。Linuxでは**iptables**コマンドを使用する

■ iptablesコマンドの書式(抜粋)

・コマンド

-A .. ルールの追加

-D .. ルールの削除

-P .. デフォルトポリシーの変更

・チェーン

FORWARD, INPUT, OUTPUT

PREROUTING, **POSTROUTING** .. IPマスカレードで使用

・ターゲット

ACCEPT .. 許可

DROP .. 破棄

REJECT .. 拒否(送信元に通知)

MASQUERADE .. 送信元IPアドレスとポート番号の変換

SNAT .. 送信元IPアドレスの変換

DNAT .. 送信先IPアドレスの変換

(次ページへ続く)



- ・ルール
- s IPアドレス .. 送信元IPアドレス
- d IPアドレス .. 送信先IPアドレス
- sport ポート番号 .. 送信元ポート番号
- dport ポート番号 .. 送信先ポート番号
- j ターゲット .. 適用されるターゲット
- p プロトコル .. プロトコルの指定(tcp, udp, icmp, all)

■ iptablesコマンドの設定例

iptables -P FORWARD DROP → FORWARDチェーンにDROPのポリシーを適用する

iptables -A INPUT -p icmp -s 192.168.1.0/24 -j ACCEPT → 送信元192.168.1.0/24からのicmpパケットを許可するルールをINPUTチェーンに追加する

iptables -A FORWARD -p tcp --dport 23 -j REJECT → 送信先ポート番号がTCP23番のパケットを拒否するルールをFORWARDチェーンに追加する

※学習ポイント: iptablesコマンドの詳細オプションを覚えるのは大変なので、設定例に挙げたようなルールについて理解できるようにする



■ アドレス変換の必要性

LAN上のプライベートIPアドレスを割り当てられたホストが、インターネット上のグローバルIPアドレスを持つホストに接続するには、ルータ/ファイアウォールでアドレス変換(NAT・IPマスカレード)が必要

■ iptablesコマンドを利用したアドレス変換の設定例

`iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE` →WAN側グローバルIPアドレスが動的に変更される場合の送信元IPアドレスの変換

`iptables -t nat -A POSTROUTING -o eth1 -j SNAT` →WAN側グローバルIPアドレスが固定の場合の送信元IPアドレスの変換

`iptables -t nat -A PREROUTING -o eth1 -j DNAT` →WAN側グローバルIPアドレスが固定の場合の送信先IPアドレスの変換

※学習のポイント:グローバルIPが動的割り当ての場合と固定の場合の違い、送信元変換と送信先変換の場合の違いについて押さえる



■SSHの実装

パッケージ名: OpenSSH

デーモン名: sshd

設定ファイル: /etc/ssh/sshd_config

■sshd_configの設定例(抜粋)

```
Port 22 #sshで使うポート番号
Protocol 2 #sshのバージョン ※バージョン1には脆弱性あり
PermitRootLogin YES | No #rootによる直接ログインの有効/無効
X11Forwarding YES | No #X11フォワーディングの有効/無効
```

■公開鍵認証方式に関する設定

ssh-keygenコマンド .. 公開鍵、秘密鍵のペアを生成

ユーザの公開鍵をsshサーバの ~/.ssh/**authorized_keys**ファイルに保存する

ssh-agentデーモン .. 秘密鍵のパスフレーズを入力する手間を省く

ssh-addコマンド .. ssh-agentに秘密鍵を登録する



※学習ポイント: sshd_configの設定項目、ssh関連コマンドについて整理する



■ /procファイルシステム

メモリにロードされているカーネルの状態にアクセスするための特殊なファイルシステム

■ /proc下の主なファイル

| | |
|------------------|----------------------|
| /proc/bus/usb | .. USBデバイスに関する情報 |
| /proc/cmdline | .. 起動時にカーネルに渡された引数 |
| /proc/cpuinfo | .. CPUに関する各種情報 |
| /proc/filesystem | .. ファイルシステムに関する情報 |
| /proc/interrupts | .. IRQごとの割り込み回数 |
| /proc/modules | .. ロードされているカーネルモジュール |
| /proc/mounts | .. マウントされているファイルシステム |
| /proc/partitions | .. パーティション情報 |
| /proc/swaps | .. 使用中のスワップ領域 |

※学習ポイント: /procファイルシステムに関しては、よく入力問題が出題される



- トラブルシューティングに役立つファイル/ディレクトリ
 - /etc/profile .. 全ユーザ共通のログインシェル設定
 - ~/.bash_profile .. 各ユーザ個別のログインシェル設定
 - /etc/bashrc .. 全ユーザ共通のbash初期化ファイル
 - ~/.bashrc .. 各ユーザ個別のbash初期化ファイル
 - /etc/login.defs .. シャドウパスワード機能の設定ファイル
 - /etc/rc.local .. ホスト固有の起動時の初期化ファイル
 - /etc/shells .. 利用可能なシェルのパスを記述するファイル

※学習ポイント: 重要な設定ファイルについては、入力問題が出題されるので、スペルを確実に覚えておく

これで、本講座のコンテンツは終了になります。ご静聴ありがとうございました。以降は、質疑応答の時間とさせていただきます。