# Born2beRoot

レビュー用資料

by terabu

# 目次

# 概要

(Project overview)

# 仮想マシンについて

＜仮想マシンとは＞

- 1 台のコンピュータ内に擬似的に再現したコンピュータ
- 物理的なリソース(CPU、メモリ、ディスクなど)を論理的に統括・分割して構築される

＜仮想マシンの利点＞

- 複数の OS を扱える
- 効率的なリソース活用

# Linux について

＜ Linux とは＞

- OS の一つ
- 無料で使えるオープンソース

＜ディストリビューションとは＞

- Linux のアプリケーションやライブラリをひとまとめにして、PC にインストールすれば使える状態にした配布物
- Debian、Ubuntu、Rocky、CentOS など

# Debian と Rocky の違い

| 項目 | Debian | Rocky |
| --- | --- | --- |
| 先祖 OS | - (Debian が初代) | RHEL(Red Hat Enterprise Linux) |
| リリース年 | 1993 年 | 2021 年 |
| 用途 | 汎用 | 主に商用 |
| PKG 管理 | apt | dnf(yum) |
| 費用 | 無料 | 無料 |
| サポート | 充実 | 充実 |

# パッケージ管理について１

＜パッケージとは＞

- ソフトウェアの実行に必要なファイルをまとめたもの
- 下記のようなもの
  - 実行ファイル
  - ライブラリ
  - 設定ファイル
  - リソース(画像・音楽ファイル)

# パッケージ管理について 2

＜パッケージ管理とは＞

- パッケージのインストール/アンインストールを管理する
- パッケージをリポジトリから自動で探す
  - リポジトリ：パッケージの保管場所 (配布サイト)
- パッケージの依存関係を自動で解決
  - 依存関係：パッケージ A を利用するために、パッケージ B をインストールする必要がある状態
- Linux では apt、yum、rpm などがある

# aptitude と apt の違い

| 項目 | apt-get | aptitude | apt |
| --- | --- | --- | --- |
| 違い | 設計にミスあり | 外部プロジェクトとして生まれた GUI をもったもの(未完成) | apt-get の設計上のミスを修正 |
| 使用推奨 | X | X | ○ |

# APPArmor

- セキュリティ機能の一つ
- 強制アクセス制御
- 外部・内部からの脅威に対して OS やアプリケーションを防護
- 各プログラムにセキュリティプロファイルを結びつけ、プログラムのできることに制限をかける
- プロファイルは、ネットワークアクセス、ファイルへの読み書き実行などの機能を制限できる

# 簡易設定

(Simple setup)

# 仮想マシン起動

＜確認項目＞

- グラフィック環境がないこと
- 接続時にパスワードが要求されること
- ユーザーで接続すること(root ではない)
- パスワードが規則通りであること

＜ OS 確認(Debian であることを確認)＞

```
$ uname -a
```

# サービス起動確認

## < UFW >

```
$ sudo ufw status
```

## < SSH >

```
$ sudo systemctl status ssh
```

# ユーザー設定

(User)

# ユーザー確認

＜ユーザー名確認＞

```
$ id -un
```

＜所属グループ確認＞

```
$ groups // (getent group sudo user42)
```

# パスワードポリシー確認

＜やること＞

1. 新規ユーザー作成
2. パスワード登録
3. 設定ファイルの確認
    - common-password
    - login.defs

# 新規ユーザー作成

＜ユーザー作成コマンド＞

```
$ sudo adduser [new_username]
```

# パスワード登録

＜確認パターン(NG)＞

- abcdeABCDE // 数字が足りない
- abcde12345 // 英大文字が足りない
- ABCDE12345 // 英小文字が足りない
- Born2beRo // 10 文字より少ない
- Born2bbbeRo // 同じ文字を 3 回繰り返している
- XXXXXXAB12 // ユーザー名が含まれている
- XXXXXXXX // 古いパスワードにない文字が 6 文字以下(設定で確認)

# パスワード登録

＜確認パターン＞
(OK)
Born2beRoot

# 設定ファイル(ポリシー)確認

```
$ sudo nano /etc/pam.d/common-password
```

- minlen：最低文字数
- lcredit：アルファベットの小文字の最低文字数(負数で指定)
- ucredit：アルファベットの大文字の最低文字数(負数で指定)
- dcredit：数字の最低文字数(負数で指定)
- maxrepeat：同一文字の連続繰り返し最大文字数
- usercheck：ユーザーが含まれているかどうか確認(0 以外を指定)
- difok：古いパスワードとは異なる必要がある最小文字数
- enforce_for_root：root にも同ポリシーを適用する

# 設定ファイル(有効期限)確認

```
$ sudo nano /etc/login.defs
```

- PASS_MAX_DAYS：パスワードの最大有効日数
- PASS_MIN_DAYS：パスワード変更の最短日数
- PASS_WARN_AGE：パスワード有効期限の警告通知日(何日前に警告するか)

＜確認＞

```
$ sudo chage -l [username]
```

# グループ割当て

＜グループ割当て＞

```
$ sudo groupadd evaluating // 作成
$ sudo usermod -aG evaluating [new_username] // 割当て
```

＜確認＞

```
$ id [new_username] -Gn
```

# パスワードポリシーのメリデメ

＜メリット＞

- パスワードの推測が困難になる
  - ID 乗っ取り対策向上

＜デメリット＞

- 管理者及びユーザーの手間が増える
  - 管理者：ポリシーの設定作業
  - ユーザー：ポリシー通りの設定、定期的なパスワード変更作業

# ホスト名&パーティション

(Hostname and partitions)

# ホスト名の確認と変更

＜確認＞

```
$ hostnamectl
```

＜変更＞

```
$ hostnamectl set-hostname [new_hostname]
```

# パーティションの確認

＜パーティションとは＞

- ハードディスクの記憶領域を論理的に分割した領域のこと

＜パーティション確認＞

```
$ lsblk
```

# LVM

＜ LVM とは＞

- ディスク管理機能
- 複数のハードディスクやパーティションにまたがった記憶領域を一つの論理ボリューム(LV)にまとめて扱うことができる
- システムを停止せずに論理ボリュームの拡大・縮小を行える

# sudo 設定

(SUDO)

# sudo について

＜ sudo とは＞

- 現在ログインしているユーザーとは別のユーザーの権限でプログラムを実行するもの
- 一部のプログラムを一般ユーザーに管理者権限で実行させたい場合に利用される

# sudo 確認

＜インストール確認＞

```
$ sudo --version
```

＜ sudo に割当て＞

```
$ sudo usermod -aG sudo [new_username] // 割当て
$ id [new_username] -Gn // 確認
```

# sudo の操作例

＜ sudo なし ＞

```
nano /etc/hosts
```

※開けない、もしくは readonly で開かれる

＜ sudo あり ＞

```
sudo nano /etc/hosts
```

※書き込み可で開ける

# sudo 設定確認

＜設定ファイル＞

```
$ sudo visudo // sudo nano /etc/sudoers
```

＜確認事項＞

1. パスワード入力制御&失敗時メッセージ
2. ログファイル
3. TTY モード
4. 使用パスの制限

# sudo 設定 パスワード入力制御&失敗時メッセージ

＜確認事項(sudoers の設定項目)＞

- パスワードのリトライが 3 回まで(passwd_tries)
- 失敗時に指定したメッセージが表示されるか(badpass_message)

# sudo 設定 ログファイル

＜確認事項(sudoers の設定項目)＞

- ログファイルの PATH が"/var/log/sudo"(logfile/iolog_dir)

【入力】

```
$ sudo less /var/log/sudo/sudo.log
```

【出力】

```
$ sudo sudoreplay -d /var/log/sudo 【ログ番号】
```

# sudo 設定 TTY モード

＜確認事項(sudoers の設定項目)＞

- TTY モードが有効になっている(requiretty)
  - 有効にすることで cron 等からの実行を許可させない

＜ TTY とは＞

- 接続端末のデバイスファイル名
- tty コマンドで表示できる

# sudo 設定 使用パスの制限

＜確認事項(sudoers の設定項目)＞

- パスが制限されていること(secure_path)

＜補足＞

- sudo は実行時に環境変数 PATH を secure_path に指定されたパスで初期化する
- 設定することにより意図しないコマンドの実行を防ぐ可能性が UP

# UFW 設定

(UFW)

# UFW について

＜ FW とは＞

- 外部アクセスの制御を行う仕組み
- 「 外部からの接続は受け付けない」「ssh だけは許す」のような設定ができる
  - 不正なアクセスを防げる可能性が UP
- UFW は FW を簡単に設定できるツール

# UFW の設定内容

```
$ sudo ufw status
```

＜確認事項＞

- 正常にインストールされていること
- ポートが 4242 のみ許可されていること

＜ポートとは＞

- 端末の接続口

# UFW ルールの追加・削除

＜ポートの追加＞

```
$ sudo ufw allow 8080 //ポート8080を追加
$ sudo ufw status // 確認
```

＜ポートの削除＞

```
$ sudo ufw delete allow 8080 // ポート8080のallowを削除
$ sudo ufw status // 確認
```

# ssh 設定

(SSH)

# SSH について

＜ SSH とは＞

- Secure Shell（セキュアシェル）の略称
- リモートコンピュータと通信するための仕組み

＜インストール確認 (ステータスの確認)＞

```
$ sudo systemctl status ssh
```

# SSH 設定確認

＜設定ファイル＞

```
$ sudo nano /etc/ssh/sshd_config
```

＜確認事項＞

- ポートが 4242 のみ使用されていること

# SSH 操作確認

ホスト側でユーザー ID でアクセス

```
$ ssh your_username@localhost -p 4242
```

アクセスできることを確認

＜他確認事項＞

- 異なるポートでアクセス　→ アクセスできないこと
- root ユーザーでアクセス　→ アクセスできないこと

# モニタリングスクリプト

(Script monitoring)

# モニタリングスクリプトの概要

- 詳細は課題参照
- bash で作成
- 10 分ごとにログインしている全ての端末に表示
  - 全ての端末に表示 → `wall` コマンドを利用
  - 10 分ごと → cron を利用

# script 出力内容の確認 1

↓各種項目は以下のコマンド・ファイルで取得できる

- OS の構成及びカーネルのバージョン
  - `uname -a` コマンド
- 物理プロセッサの数
  - 「/proc/cpuinfo」ファイル："physical id"
- 仮想プロセッサの数
  - 「/proc/cpuinfo」ファイル："processor"

# script 出力内容の確認 2

- サーバー上で現在使用可能なメモリとその使用率(%表記)
  - `free` コマンド："Mem" の行(total/used)
- サーバー上で現在使用可能なディスクとその使用率(%表記)
  - `df -BM -T --total` コマンド：”total”の行(1M-blocks/Used/Use%)
- 現在のプロセッサの使用率(%表記)
  - `top -bn1` コマンド：”load average”
- 前回再起動の日時
  - `who -b` コマンド

# script 出力内容の確認 3

- LVM がアクティブかどうか
  - `lsblk` コマンド：”lvm”の記述があれば LVM が利用されている
- アクティブな接続の数
  - `netstat` コマンド："ESTABLISHED"の数
- サーバーを使用しているユーザーの数
  - `who` コマンド：ユーザー数(重複除外)

# script 出力内容の確認 4

- サーバーの IPv4 アドレスとその MAC アドレス
  - IPv4 アドレス
    - `hostname -I` コマンド
  - MAC アドレス
    - `ip a` コマンド：”link/ether”
- sudo プログラムで実行されたコマンドの数
  - 「/var/log/sudo/sudo.log」ファイル：”COMMAND”の数

# cron について

- プログラムを定期的に自動実行させるための仕組み
- 設定は crontab ファイルで行う

```
$ sudo crontab -l
```

- crontab の基本書式
  - ”分　時　日　月　曜日　実行スクリプト名"
  - ”*” は何も指定しない意味
  - ”/n”のように書くと n おきに実行する

# cron 検証

＜ cron の変更＞

```
$ crontab -u root -e
```

※指示の通り編集する

＜ cron の停止＞

```
$ sudo sytemctl stop cron // cron停止
$ sudo sytemctl disable cron // cronの自動起動無効化
```

# 終わり

ありがとうございました！