01_Linuxサーバデータベース - Redmineのデータベースの日次バックアップ。(MySQL：暗号化付与)

こちらの記事で挙げたRedmineなどのMySQLを実行するスクリプト。

https://atelier.reisalin.com/projects/zettel/knowledgebase/articles/40

この問題点を修正します。

問題点

- むきだしのSQLファイルが平文で格納されてしまうのはセキュリティ的によろしくありません。
- MySQLのバックアップ時に使うアカウントファイルが誰でも読み取れるのも問題です。

そこで、バックアップされたファイルにパスワードをかけることで簡単な防波堤を作ることにします。

前提

上記URLに併せます。

1. MySQL dumpを行うDBにRELOAD権限があること。
2. 次の環境で動作を確認しています。
    - Ubuntu 20.04
    - MySQL 8.0.32

実施した手順

さっくりとした手順

1. バックアップディレクトリを作成します。
2. DBにアクセスするためのアカウント情報を記したファイルを作成します。
3. 開封パスワードを格納するディレクトリを作成します。
4. バックアップスクリプトを作成します。
5. crontabに登録します。

バックアップディレクトリを作成します。

```
sudo　mkd -p　/home/backup/mysql
#　運用に合わせて指定ください。ファイルサーバや別パーティションにマウントしている方がサーバ事態の障害発生でも冗長化を持たせられます。

sudo　chow -R　hoge:hoge　/home/backup/mysql
#　ディレクトリの所有者をログインユーザに修正します

cd　/home/backup/mysq&& pwd
#　指定したディレクトリに移動します
```

DBにアクセスするためのアカウントファイルを作成します。

Cronによる自動実行を前提としているため、スクリプト実行時にDBユーザとパスワードを記したファイルを読み込むことでセキュリティのリスクを抑えます。

```
sudo　mkd -p　/home/hoge/db_password
#　運用に合わせて指定ください。

cd　/home/hoge/db_passworc&& pwd
#　指定したディレクトリに移動します
```

以下の内容を教義・信仰に沿ったエディタで作成します。(【】内は取り除き、自分の設定に合わせます)

- アカウントファイル内容
    - ファイル名:account.txt

```
[client]
user　=　【RedmineのDBユーザ】
```

password = "【RedmineのDBユーザ用パスワード】"

その後、このファイルの読み取り権限を変更します。

chmod 400 account.txt

ls -l account.txt
パーミッションが400であることを確認します

アカウントファイルでアクセスできることを確認

mysql --defaults-extra-file=【アカウントファイルを格納したディレクトリ】/account.txt

#MySQLのプロンプトが出れば成功です。exitで抜けます。

スクリプト作成

以下の内容を教義・信仰に沿ったエディタで作成します。

- スクリプト内容
  - スクリプト名:pw_mysql_daily_backup.sh

```
#!/bin/bash

##  変数ここから  ##
#  SQLをバックアップするディレクトリ(保管先)を指定します。運用に合わせて指定ください。
backup_dir="/home/backup/mysql"
#  保持するバックアップの世代を日数で指定します。
keep_days=7
#  ファイルに付与する日付/作業ディレクトリ名/バックアップファイル名を指定します。
current_date=$(date  +%Y%m%)l
backup_name="redmine_mysql_${current_date}"
zip_file="redmine_mysql.${current_date}.zip"
#  アカウントファイルを指定します。運用に合わせて指定ください。
credentials_file="$HOME/redmine/account.txt"
#  パスワードを記録するファイル名を指定します。運用に併せてして指定ください。
password_dir="$HOME/restore_redmine"
password_file="${password_dir}/mysql-restore.$current_date.txt"
#  redmineのデータベース名を指定します。
database_name=redmine
#  バックアップ時に指定するオプションを指定します。
options="--defaults-extra-file=$credentials_file  --no-tablespaces  --single-transaction"
##  変数ここまで  ##

##  処理ここから  ##

#  1.アカウントファイルのパーミッションが400かどうかチェックします。
#  400以外は処理そのものを終了します。
permissions=$(stat -c "%a" "$credentials_file")
if [ "$permissions" != "400" ]; then
    echo "アカウントファイルのパーミッションは400である必要があります。"
    exit 1
fi

#  2.一時的なバックアップディレクトリを作成します。
mkdir "${backup_dir}/${backup_name}"

#  3.  mysqldumpを実行してデータベースのバックアップを取ります。
mysqldump $options -h  localhos$database_name > "${backup_dir}/${backup_name}/${backup_name}.sql"

#  4.  パスワードによる暗号化を実施します。
password=$(openssl  ranc-base64  1)!
cd "${backup_dir}/${backup_name}"
zip -r "${backup_dir}/${zip_file}" -P "$password" .
cd  -

#  5.  一時的なバックアップディレクトリを削除します。
```

```
rm -rf "${backup_dir}/${backup_name}"

#  6.  解凍パスワードを指定ディレクトリに保存します。
echo $password > $password_file

#  7.パスワードの読み取り権限を600に変更します。
chmod 600 $password_file

#  8.  保持期間より古いバックアップファイルを削除します。
find "$backup_dir" -name "redmine_mysql.*.zip" ! -type f -newermt "${keep_days} days ag -delete
find "$password_dir" -name "*restore*.txt" ! -type f -newermt "${keep_days} days ag -delete

##  処理ここまで
```

前回との修正点

1. 変数と処理のセクションを明確化しています。
2. アカウントファイルのパーミッションチェックを行い、400以外は処理を中止します。
3. openssslで生成したパスワードで暗号化します。（このパスワードはランダムで生成されるので運用者は覚える必要がありません）
4. 圧縮と同時に暗号化を行うので、gz形式からzip形式に変更しています。
5. このパスワードを任意のディレクトリに転送します。

- 実行権限の付与

```
chmod +x pw_mysql_daily_backup.sh
```

動作確認

```
cd 【スクリプトを格納したディレクトリ】&& pwd
bash pw_mysql_daily_backup.sh
```

以下を確認します。

1. エラーなく実行できること
2. バックアップ格納ディレクトリにredmine.sql.実行日付.zip形式でファイルが作成されること
3. パスワードファイル格納ディレクトリにファイル名.実行日付.txt形式でファイルが作成されること
4. unzip redmine.sql.実行日付.zipでファイル解凍時にパスワードを確認されること
5. パスワードファイルで暗号化されたファイルを解凍することができること

Crontab設定

Cron登録

```
crontab -e
```

登録内容例

```
0 0 * * * /home/backup/mysql/pw_mysql_daily_backup.sh
#  実行時刻、頻度などは自分の運用形態に合わせます。
#  また、既に平文でのバックアップスクリプトを設定している場合はコメントアウトして処理を外します。
```

Cron登録確認

```
sudo ta -20 /var/log/cron.log
```

操作時刻に

- BEGIN EDIT
- REPLACE
- END EDIT

が表示されれば設定は完了です。

動作確認日

2023/02/18

ファイル

Redmine運用-2-.jpeg	207 KB	2024/01/05	手動人形