セキュリティとアクセス権限管理

# MySQLにおけるセキュリティとは

MySQLにおけるセキュリティとは、MySQLが守備範囲にしているセキュリティと、その範囲外であるセキュリティの２つに分けられます。

MySQLのドキュメントにはセキュリティのガイドラインがあります。

ドキュメントの概要（一部）は以下のとおりです。

１．rootユーザ以外にmysqlデータベースのuserテーブルへのアクセス権限を付与しない

**※MySQLではユーザ管理情報をmysqlデータベースのuserテーブルで管理している。**

２．MySQLのアクセス権限の仕組みを学習する

**・接続元がどこであっても許可する設定を利用しない**

・GRANT、REVOKE文で権限を管理する

**・与える権限は最小にする**

３．平文のパスワードを利用しない。SHA2などハッシュを利用する

４．パスワード解析を避けるために辞書にある単語を使わない

なお、詳細についてはリンク先を参照してください

https://dev.mysql.com/doc/refman/8.0/en/security-guidelines.html

データベース演習IIでは、**太字にしている部分を**学習します。

ハンズオン　MySQLのユーザ管理情報を確認する

1. 管理者権限でMySQLにログインをする。
2. mysql.userテーブルを参照して、データベースのユーザ情報を確認する。

SELECT USER, HOST FROM MYSQL.USER;

1. 一般ユーザでMySQLにログインする

mysql –u dbuser –p

1. mysql.userテーブルを参照して、データベースのユーザ情報を確認する。

SELECT USER, HOST FROM MYSQL.USER;

# ユーザごとのアクセス権限管理

アクセス権限管理とは、データベースのユーザと接続元の組合せに対して操作対象・内容の許可を与える仕組みです。先程、mysql.userテーブルを確認しましたが、以下のようなデータが格納されていると思います。USER列にはデータベースユーザがHOST列には接続元の情報が管理されています。

|  |  |
| --- | --- |
| USER | HOST |
| dbuser  mysql.infoschema  mysql.session  mysql.sys  root | %  localhost  localhost  localhost  localhost |

※％はワイルドカード

dbuserユーザはホストが％(ワイルドカード)のみなので、どの接続元からでもMySQLにアクセスすることができます。しかし、rootユーザはホストがlocalhostに指定されているため、localhost（自サーバ）からのみアクセス可能で他の接続元からログインすることができません。

接続元(192.0.2.0)

接続元(198.51.100.0)

データベースサーバ

(localhost)

MySQL

サーバ

(mysqld)

TCP/IP

クライアント

**rootユーザ**

)

クライアント

**rootユーザ**

クライアント

**dbuserユーザ**

# 接続元の設定方法

接続元の設定は、ユーザを作成するCREATE USER文に情報を記述します。

書式：CREATE USER ユーザ名**@接続元** ;

IDENTIFIED WITH MYSQL\_NATIVE\_PASSWORD BY パスワード ;

※接続元を省略すると、%が設定されます。

既存のユーザの接続元を変更する場合は、RENAME文を使用します。

書式：RENAME USER ユーザ名@接続元 TO ユーザ名**@新接続元** ;

この接続元を適切に設定することで、「**接続元がどこであっても許可する設定を利用しない**」セキュリティの向上につながります。

ハンズオン　ユーザの接続元を管理する

１．管理者ユーザでMySQLにログインをする

２．ダミーの接続元を指定した、テストユーザを作成します。

CREATE USER [dbtest@'192.0.2.0](mailto:dbtest@'192.0.2.0)'

IDENTIFIED WITH MYSQL\_NATIVE\_PASSWORD BY 'ecc' ;

３．mysql.userテーブルを参照して、データベースのユーザ情報を確認する。

SELECT USER, HOST FROM MYSQL.USER;

４．新しいコマンドプロンプトでテストユーザでのログインを試してみる。

mysql -u dbtest –p

５．管理者ユーザで、テストユーザの接続元をローカルホストに変更する。

RENAME USER [dbtest@'192.0.2.0](mailto:dbtest@'192.0.2.0)' TO dbtest@'localhost' ;

SELECT USER, HOST FROM MYSQL.USER;

６．テストユーザでMySQLにログインをする。

mysql -u dbtest –p

# 権限の設定方法

ユーザを作成した時点では、どのデータベースにも接続できません。

権限を付与することにより初めてデータベースにアクセスすることができます。

権限の付与と剥奪はデータベース演習Iでも学習したGRANTとREVOKEです。

・権限の付与

書式：GRANT 権限の種類 ON 指定対象TO ユーザ名@接続元

・権限の剥奪

書式：REVOKE 権限の種類 ON 指定対象FROM ユーザ名@接続元

ただし、信頼しているサーバだからといって、あらゆる情報にアクセス出来るとセキュリティが低下しますので「**与える権限は最小にする**」ことを忘れないでください。

接続元

データベースサーバ

MySQLサーバ(mysqld)

クライアント

**dbuserユーザ**

)

mysql

studb

SELECT

SELECT

権限には付与可能な対象として、全体、データベース、表、列などがあります。

例えば、UPDATE権限は表だけでなく、列に対して権限を付与することが出来るので特定の列のみ更新可能など、細やかな設定をすることが出来ます。

・権限の一覧（一部抜粋）

|  |  |  |
| --- | --- | --- |
| 付与レベル | 権限 | 説明 |
|  | ALL [PRIVILEGES] | GRANT OPTIONとPROXY以外のすべての権限 |
|  | USAGE | 権限なし |
| 全体 | CREATE USER | CREATE USER、DROP USER、RENAME USER、REVODE ALL PRIVILEGESの実行権限 |
| 全体  データベース  表 | CREATE | データベースやテーブルの作成権限 |
| DROP | データベースやテーブルの削除権限 |
| ALTER | データベースやテーブルの変更権限 |
| DELETE | DELETE命令の実行権限 |
| 全体  データベース  表  列 | SELECT | SELECT命令の実行権限 |
| INSERT | INSERT命令の実行権限 |
| UPDATE | UPDATE命令の実行権限 |

ハンズオン　ユーザの権限を管理する

１．管理者ユーザでMySQLにログインを行う。

２．テストユーザの権限を確認します。

SHOW GRANTS FOR dbtest@'localhost' ;

|  |
| --- |
| Grants for dbtest@localhost |
| GRANT USAGE ON \*.\* TO `dbtest`@`localhost` |

※全ての領域に対して権限を持っていない。

３．テストユーザに対して全ての領域に対して、全ての権限を付与する。

**※本来は、rootと同等の権限は付与するべきではないです**

GRANT ALL ON \*.\* TO dbtest@'localhost' ;

SHOW GRANTS FOR dbtest@'localhost' ;

４．新しいコマンドプロンプトでMySQLにテストユーザでログインする

mysql -u dbtest –p

５．studbにあるSTORE表の情報を表示する

SELECT \* FROM STUDB.STORE ;

６．mysql.userテーブルを参照して、データベースのユーザ情報を確認する。

SELECT USER, HOST FROM MYSQL.USER;

７．**管理者ユーザ**で全ての権限の剥奪を行う

REVOKE ALL ON \*.\* FROM dbtest@'localhost' ;

SHOW GRANTS FOR dbtest@'localhost' ;

# rootユーザのパスワードのリセット

rootパスワードを忘れてしまったり、設定したはずが正しく設定できていないため、ログインできなくなることがあります。このような状況に対処できるように、rootパスワードのリセット方法を学習します。

MySQLには認証の無効化をする、skip-grant-tablesオプションが存在します。このオプションを利用することで**パスワードを入力しなくてもMySQLにログインすることが出来ます**。ただし、認証の無効化は大変危険な状態になりますので、この設定を行った場合、**ネットワークからの接続を受付けない設定も自動的に設定されます**。

ここで、一つの問題点が発生します。デフォルトでは、localhostであってもTCP/IP経由でMySQLに接続を行います。その為、skip-grant-tablesオプションを設定するとネットワークからの接続が出来ない為、MySQLにそもそもログイン出来なくなります。

そこで、さらにソケット経由で直接MySQLに接続するオプションを追加して認証なしでMySQLに接続を行い、rootパスワードの変更を行います。

データベースサーバ

MySQL

サーバ

(mysqld)

TCP/IP

クライアント

**rootユーザ**

クライアント

**rootユーザ**

直接、接続する

必要がある

・rootパスワードリセットに関連するオプション

|  |  |  |
| --- | --- | --- |
| グループ | オプション | 説明 |
| mysqld | skip-grant-tables | 認証の無効化。サーバへのアクセス権を持つすべてのユーザにすべてのデータベースに対する無制限アクセス権が与えられる。 |
| mysqld | socket=MYSQL | Unixプラットフォームでは、この変数はローカルクライアント接続に使用されるソケットファイルの名前です。デフォルトは /tmp/mysql.sock です。Windowsでは、この変数はローカルクライアント接続に使用される名前付きパイプの名前です。デフォルト値はMySQL |
| mysqld | enable-named-pipe | 名前付きパイプのサポートを有効にします。このオプションは Windows にのみ適用されます。 |

ハンズオン　rootパスワードをリセットする

１．データベースの停止を行う。

２．my.iniファイルをエディタで開いて、オプション情報を編集する。

|  |  |  |
| --- | --- | --- |
| グループ | 変更前 | 変更後 |
| mysqld | # enable-named-pipe | enable-named-pipe |
| mysqld | # socket=MYSQL | socket=MYSQL |
| mysqld |  | skip-grant-tables |

※skip-grant-tablesは新規追加の為、mysqldグループの一番下に記述でOKです。

３．データベースの起動を行う。

４．名前付きパイプを使い認証無しでMySQLにログインを行う

mysql -u root --pipe

５．rootユーザの新しいパスワードを設定する。

FLUSH PRIVILEGES;

ALTER USER root@'localhost' IDENTIFIED BY 'admin';

６．データベースの停止を行う。

７．my.iniファイルをエディタで開いて、オプション情報を元に戻す。

|  |  |  |
| --- | --- | --- |
| グループ | 変更前 | 変更後 |
| mysqld | enable-named-pipe | # enable-named-pipe |
| mysqld | socket=MYSQL | # socket=MYSQL |
| mysqld | skip-grant-tables | # skip-grant-tables |

８．データベースの起動を行う。

９．新しいパスワードでrootユーザのログインを行う。

mysql -u root -p