MySQL 04

2023.06

DB보안

- 루트 계정을 일반 업무에 사용하면 해킹당할 경우, 통제 불능 상태가 되므로, 루트 계정은 로컬에서만 사용하도록 하고 일반 업무는 일반 사용자에게 권한을 부여하도록 합니다.

- 일반 사용자 계정을 생성하고, 특정 업무를 수행하기 위해 권한을 최소한으로 부여하는 것을 Least Privilege(최소한의 특권)이라고 합니다.

- 루트 계정을 원격에서 사용하도록 설정하면, DB취약점 공격에 성공할 경우, 공격자에게 루트 권한이 주어지게 됩니다. 루트 권한은 모든 권한이 가능하므로 공격자가 DB를 통째로 복사 및 삭제할 수 있게 되므로 매우 위험합니다.

- MySQL과 MariaDB에서 보안을 강화하기 위한 명령어

mysql_secure_installation

새 사용자 만들기

* web이라는 database를 생성해 봅니다.
create database web;

* twice라는 사용자를 만들고 비밀번호는 baby로 설정합니다.
create user twice@localhost identified by 'baby';

* web이라는 db에 모든 권한 부여합니다.
grant all privileges on web.* to twice@localhost;

* 위에서 설정한 내용을 즉시 적용합니다.
flush privileges;
exit

* twice로 로그인합니다.
mysql -u twice -p
use web;

회원 테이블

* 아래와 같이 만들어 봅니다.
  테이블 : users
  컬럼 : id, passwd, name 생성 (셋 모두 varchar(20)으로 설정)

```
create table users(
  id varchar(20) not null primary key,
  passwd varchar(20),
  name varchar(20)
  );

insert into users values ('twice', 'baby', 'dahyun');
        // 패스워드를 평문으로 저장하면 안됨 (개인정보 보호법 위반!!!)
        // 개인정보를 암호화해서 처리, 특히 비밀번호는 일방향 암호화로 저장)

delete from users where id='twice';
```

패스워드를 암호화해서 저장하는 방법

* 패스워드를 저장할 때에는 password( )함수를 사용해서 저장

```
insert into users values ('twice', password('baby'), 'dahyun');          // 에러 발생!!!
```

-----> password( ) 연산 결과가 저장되는데 varchar(20)을 초과하기 때문에 저장할 수 없음

```
alter table users modify column passwd varchar(300);

insert into users values ('twice', password('baby'), 'dahyun');
insert into users values ('blackpink', password('princess'), 'jisu');
insert into users values ('itzy', password('rainbow'), 'yuna');

select * from users;
```

-----> 나온 값을 복사해서 crackstation.net 에 넣고 확인

* 대응방법

- 패스워드를 Dictionary에 없는 것을 사용해야 함
- 대/소/숫/특 섞어서 12자리 이상

안전한 패스워드 저장방법

insert into users values ('ive', password('elEven12#$'), 'yujin');
------> SHA1이지만 크래킹 안됨 (Dictionary에 없기 때문)

insert into users values ('bravegirls', sha2('rolling',256), 'yuna');
------> SHA2(256bit)를 사용하고 있지만, 쉬운 패스워드를 사용하기 때문에 쉽게 크래킹됨

* 안전한 방법은 아래와 같이 2가지를 모두 적용해야 함
1) 안전한 해쉬함수를 사용한다.
2) 패스워드를 복잡하게 설정한다.
3) 주기적으로 패스워드를 변경한다. (특히 직원이 퇴사, 이직 등)

select * from users;
→ hash값이 나오면 crackstation.net 에 입력하고 확인해봅니다.

해쉬 함수 (Hash Function)

* 해쉬함수의 원리 : 원문(파일,텍스트)을 해쉬함수에 넣으면 해쉬값(hash code)이 출력됨

1. 고정길이 출력
- 원문(동영상, 문서, 이미지)의 길이와 관계없이 항상 고정된 길이로 출력됨
  MD5(128bit), SHA1(160bit), SHA2-256(256bit)

2. 일방향 함수
- 해쉬함수와 해쉬값을 알아도 원문으로 복구를 할 수 없음 (역연산 불가능)
  동영상이 2G, MP3가 7M, 문서 2M ----> MD5 ----> 128bit(해쉬값)

3. 충돌 방지
- 충돌 : 원문이 다른데 해쉬값이 같은 경우 -----> 원문을 몰라도 원문과 동일한 효과를 볼 수 있음
- 충돌될 가능성 : MD5 ---> 1/(2의128승) == 0, SHA1(160bit) ---> 1/(2의160승) == 0
  (128bit = 2의 128승 == 사하라 사막의 모래알 개수)
- 컴퓨터의 발전으로 MD5와 SHA1의 충돌쌍을 찾기도 함 -----> 안전성에 문제 생김 ----> MD5,SHA1 사용금지 권고

파일로 내보내기

1) Databases 전체 내보내기
mysqldump -u[아이디] -p[패스워드] -all-databases > 저장될 파일명
ex) mysqldump -uroot -ppass -all-databases > test.sql

계정은 root,
패스워드는 pass
라고 가정합니다.

2) Database만 내보내기
mysqldump -u[아이디] -p[패스워드] [디비명] > 저장될 파일명
ex) mysqldump -uroot -ppass test > test.sql

3) 테이블만 내보내기
mysqldump -u[아이디] -p[패스워드] [디비명] [테이블명] > 저장될 파일명
ex) mysqldump -uroot -ppass test student > test.sql

4) 테이블구조만 내보내기
mysqldump -u[아이디] -p[패스워드] -no-data [디비명] [테이블명] > 저장될 파일명
ex) mysqldump -uroot -ppass -no-data test student > test.sql

5) 테이블구조를 제외한 데이터만 내보내기
mysqldump -u[아이디] -p[패스워드] -no-create [디비명] [테이블명] > 저장될 파일명
ex) mysqldump -uroot -ppass -no-create test student > test.sql

sql 파일 실행하기

* 운영체제에서 SQL 파일을 DB에 넣기
mysql -u[아이디]-p[패스워드] [디비명]< 파일명

<mysqldump를 이용하지 않은 방법>
mysql -u[아이디] -p[패스워드] database > 저장될 파일명
mysql -u[아이디] -p[패스워드] database < 들여올 파일명
mysql -u[아이디] -p[패스워드] < 실행할 파일명

* MySQL(MariaDB) 내에서 외부의 파일을 실행하기
1) 리눅스에서 DB가 실행중인 경우 파일 실행하기
source /home/john/goods.sql
// SQL을 그대로 실행하기 (DB생성, Table생성, Insert수행 등등)

2) 윈도우에서 DB가 실행중인 경우 파일 실행하기
source C:/intel/goods.sql
source C:₩₩intel₩₩goods.sql

goods.sql 파일 실습

cmd> mysql -u root -p
source C:/intel/goods.sql

* 이름, 도시, 국가만 골라서 보기
select customerName,city,country from customers;

* 미국 회사만 보려면?
select customerName,city,country from customers where country = 'USA';

* 뉴욕에 있는 회사만 보려면?
select customerName,city,country from customers where city='NYC';

goods.sql 파일 실습

* 직원 테이블 전체 보기

```
select * from employees
```

* 이름,성,메일,직책만 보기

```
select  firstName,lastName,email,jobTitle  from  employees;
```

* orders에서 2005년것만 보려면?

```
desc  orders;                    // 테이블 구조를 보는 명령
select  *  from  orders  where  orderDate  between  '2005-01-01' and '2005-12-31' ;
```

* 주문일자가 2005년 상반기 인 것만 골라서 주문날짜와 배송시작일을 보려면

```
select orderDate,shippedDate from orders where orderDate between '2005-01-01' and '2005-06-30' ;
```

goods.sql 파일 실습

* product라는 상품 테이블에서 '비행기(Planes)'만 골라서 productVendor 리스트를 추출해보세요.

```
desc  products;                          // 테이블 구조를 파악
select  productVendor  from  products  where  productLine='Planes';
```

* 배만드는 회사는?

```
select  productVendor  from  products  where  productLine='Ships';
```

* 빈티지 카(Vintage Cars) 만드는 회사는?

```
select  productVendor  from  products  where  productLine='Vintage Cars';
```

* 중복을 제거하고 보려면?

```
select distinct productVendor from products where productLine='Vintage Cars';
```

과제

과제 목적 : 가장 많이 구입한 우수 고객 10명을 골라서 감사의 상품권을 고객의 주소로 보내려고 합니다.

1) 상세 주문 테이블의 컬럼 종류를 살펴봅니다.

desc orderdetails;

2) 주문금액이 많은 고객은 주문량(quantityOrdered)와 제품가격(priceEach)의 곱으로 계산할 수 있습니다. 따라서, 주문량과 가격을 곱해서 vip 컬럼으로 만들기

select quantityOrdered*priceEach as vip from orderdetails;

3) vip 컬럼에 따라 내림차순 정렬

4) 너무 많이 나오므로 10개만 골라서 보기

요약) 주문량과 가격을 곱해서 vip 컬럼으로 만들고, vip 컬럼을 기준으로 내림차순 정렬을 한 후, 10개만 골라서 보기

5) 누가 주문했는지 주문 번호 추출

6) 고객 번호를 추가해서 누가 많이 주문했는지 확인

7) 고객 회사이름을 추가해서 누가 많이 주문했는지 확인

8) 고객의 이름, 전화번호, 주소를 출력하기 (join)

위와 같은 방법이 아닌, 다른 방법으로도 VIP고객 10명을 생각해 보도록 합니다.

QnA