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

# Ingres SQL Injection Cheat Sheet

Ingres seems to be one of the less common database backends for web applications, so I thought it would be worth installing it and making some notes to make my next Ingres-based web app test a little easier.

Below are some tabulated notes on how to do many of thing you'd normally do via SQL injection. All tests were performed on Ingres 9.2.0 alpha Build 108 for Linux. The Ingres download page is here.

This page will probably remain a work-in-progress for some time yet. I'll update it as I learn more.

This post is part of 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.

| | |
|---|---|
| Version | select dbmsinfo('_version'); |
| Comments | SELECT 123; — comment<br>select 123; /* comment */ |

| | |
|---|---|
| Current User | select dbmsinfo('session_user');<br>select dbmsinfo('system_user'); |
| List Users | First connect to iidbdb, then:<br>SELECT name, password FROM iiuser; — or<br>SELECT own FROM iidatabase; |
| Create Users | create user testuser with password = 'testuser';– priv |
| List Password Hashes | First connect to iidbdb, then:<br>select name, password from iiuser; |
| List Privileges | select dbmsinfo('db_admin');<br>select dbmsinfo('create_table');<br>select dbmsinfo('create_procedure');<br>select dbmsinfo('security_priv');<br>select dbmsinfo('select_syscat');<br>select dbmsinfo('db_privileges');<br>select dbmsinfo('current_priv_mask'); |
| List DBA Accounts | TODO |
| Current Database | select dbmsinfo('database'); |
| List Databases | SELECT name FROM iidatabase; — connect to iidbdb |
| List Columns | select column_name, column_datatype, table_name, table_owner from iicolumns; |
| List Tables | select table_name, table_owner from iitables;<br>select relid, relowner, relloc from iirelation;<br>select relid, relowner, relloc from iirelation where relowner != '$ingres'; |
| Find Tables From Column Name | SELECT table_name, table_owner FROM iicolumns WHERE column_name = 'value' |
| Select Nth Row | Astoundingly, this doesn'tseem to be possible! This is as close as you can get:select top 10 blah from table;<br>select first 10 blah form table; |
| Select Nth Char | select substr('abc', 2, 1); — returns 'b' |
| Bitwise AND | The function "bit_and" exists, but seems hard to use. Here's an example of ANDing 3 and 5 together. The result is a "byte" type |

| | |
|---|---|
| | with value ?01:select substr(bit_and(cast(3 as byte), cast(5 as byte)),1,1); |
| ASCII Value -> Char | TODO |
| Char -> ASCII Value | TODO<br>(The “ascii” function exists, but doesn’t seem to do what I’d expect.) |
| Casting | select cast(123 as varchar);<br>select cast(’123′ as integer); |
| String Concatenation | select ‘abc’ || ‘def’; |
| If Statement | TODO |
| Case Statement | TODO |
| Avoiding Quotes | TODO |
| Time Delay | ???See Heavy Queries article for some ideas. |
| Make DNS Requests | TODO |
| Command Execution | Impossible? |
| Local File Access | TODO |
| Hostname, IP Address | SELECT dbmsinfo(‘ima_server’) |
| Location of DB files | SELECT dbdev, ckpdev, jnldev, sortdev FROM iidatabase WHERE name = ‘value’ — primary location of db<br>SELECT lname FROM iiextend WHERE dname = ‘value’ — extended location of db<br>SELECT are FROM iilocations where lname = ‘value’ – all area (ie directory) linked with a location |
| Default/System Databases | SELECT name FROM iidatabase WHERE own = ‘$ingres’ — connect to iidbdb |
| Installing Locally | The Ingres database can be downloaded for free from http://esd.ingres.com/<br>A pre-built Linux-based Ingres Database Server can be download from |

| | |
|---|---|
| | http://www.vmware.com/appliances/directory/832 |
| Database Client | TODO<br>There is a client called “sql” which can be used for local connections (at least) in the database server package above. |
| Logging in from command line | $ su - ingres<br>$ sql iidbdb<br>* select dbmsinfo(‘_version’); go |
| Identifying on the network | TODO |

The following areas are interesting enough to include on this page, but I haven’t researched them for other databases:

| Description | SQL / Comments |
|---|---|
| Batching Queries Allowed? | Not via DBI in PERL. Subsequent statements seem to get ignored: select blah from table where foo = 1; select … doesn’t matter this is ignored. |
| FROM clause mandated in SELECTs? | No. You don’t need to select form “dual” or anything. The following is legal:<br>select 1; |
| UNION supported | Yes. Nothing tricky here. The following is legal:<br>select 1 union select 2; |
| Enumerate Tables Privs | select table_name, permit_user, permit_type from iiaccess; |
| Length of a string | select length(‘abc’); — returns 3 |
| Roles and passwords | First you need to connect to iidbdb, then:<br>select roleid, rolepass from iirole; |
| List Database Procedures | First you need to connect to iidbdb, then:<br>select dbp_name, dbp_owner from iiprocedure; |
| Create Users + Granting Privs | First you need to connect to iidbdb, then:<br>create user pm with password = ‘password’;<br>grant all on current installation to pm; |

## Thanks

Pentestmonkey gratefully acknowledges the contributions of:

Jean-Pierre Zuate

Tags: cheatsheet, database, ingres, pentest, sqlinjection

Posted in SQL Injection