

# 技术分享 | mysql 客户端对配置文件的读取顺序

原创 余振兴 爱可生开源社区 2022-08-17 16:34 发表于上海

收录于合集

#mysql客户端 1 #参数默认值 login-path 1

作者：余振兴

爱可生 DBA 团队成员，热衷技术分享、编写技术文档。

本文来源：原创投稿

\* 爱可生开源社区出品，原创内容未经授权不得随意使用，转载请联系小编并注明来源。

我们都知道使用 mysql 客户端去访问 MySQL 数据库时,会以一定的顺序去读取不同位置的配置文件,但在一次做测试时,发现除了按照顺序读取默认的配置文件的默认路径外,mysql 还有额外的读取配置文件的行为,以下是在我本地测试环境做测试时遇到的一个有意思的小知识点

## 一、场景现象

在本地做测试时，发现一个奇怪的现象，当我使用 socket 打算登录数据库，发现不指定用户时，默认并不是用的 root 用户登录，而是被修改为了 zhenxing 用户

```
[root@10-186-61-162 ~]# mysql -S /data/mysql/3306/data/mysql.sock -p
Enter password:
ERROR 1045 (28000): Access denied for user 'zhenxing'@'127.0.0.1' (using password: NO)
```

通过观测当前 mysql 客户端的默认参数行为，可以看到与报错一致，默认用户确实是变为了 zhenxing

```
[root@10-186-61-162 ~]# mysql --help | egrep "user|host|port"
-h, --host=name      Connect to host.
-P, --port=#         Port number to use for connection or 0 for default to, in
-u, --user=name      User for login if not current user.
host                  127.0.0.1
port                  3306
user                  zhenxing
```

在这里作为 DBA，我们的第一反应是肯定是查看 /etc/my.cnf 文件中是否对默认用户做了配置，于是查看该配置文件的客户端配置参数，如下

```
[client]
```

```
host          = 127.0.0.1
user          = root
port          = 3306

[mysql]
host          = 127.0.0.1
user          = root
port          = 3306
prompt        = '\U[\d]> '
```

发现配置文件中的反而配置值是 root 用户，并没有对 zhenxing 用户做配置，看来读取的还不是这个配置文件，那是不是读取了其他配置文件呢，继续排查其他的配置文件

## 二、排查思路

### 1、获取配置文件读取顺序

我们先打印出所有可能读取的配置文件及其读取的顺序做逐个排查

```
## 查看mysql 客户端读取配置文件的顺序

[root@10-186-61-162 ~]# mysql --verbose --help|grep my.cnf
                order of preference, my.cnf, $MYSQL_TCP_PORT,
/etc/my.cnf /etc/mysql/my.cnf /usr/local/mysql/etc/my.cnf /data/mysql/3306/base/my.cnf ~/.my.cnf
```

### 2、排查 /etc/my.cnf

/etc/my.cnf 在前面已经确认没有做相关配置，这里直接跳过

### 3、排查 /etc/mysql/my.cnf

查看 /etc/mysql/my.cnf 配置，发现不存在相关配置，排除

```
[root@10-186-61-162 ~]# cat /etc/mysql/my.cnf
cat: /etc/mysql/my.cnf: 没有那个文件或目录
```

### 4、排查 /usr/local/mysql/etc/my.cnf

查看 /usr/local/mysql/etc/my.cnf 配置,发现不存在相关配置,排除

```
[root@10-186-61-162 ~]# cat /usr/local/mysql/etc/my.cnf
cat: /usr/local/mysql/etc/my.cnf: 没有那个文件或目录
```

## 5、排查 /data/mysql/3306/base/my.cnf

查看 `/data/mysql/3306/base/my.cnf` 配置,发现不存在相关配置,排除

```
[root@10-186-61-162 ~]# cat /data/mysql/3306/base/my.cnf
cat: /data/mysql/3306/base/my.cnf: 没有那个文件或目录
```

## 6、排查 ~/.my.cnf

查看 `~/.my.cnf` 依旧不存在相关配置,排除

```
[root@10-186-61-162 ~]# cat ~/.my.cnf
cat: /root/.my.cnf: 没有那个文件或目录
```

至此按照 `mysql --verbose --help|grep my.cnf` 获取的配置文件读取路径都被排除,都未对用户 zhenxing 做配置

## 7、使用 no-defaults 排除配置文件干扰

- 尝试用 `--no-defaults` 不读取任何配置文件排除配置文件的干扰,看是否会恢复正常

```
[root@10-186-61-162 ~]# mysql --help|grep no-defaults
--no-defaults          Don't read default options from any option file

## 查看不读取配置文件时,客户端的默认值
[root@10-186-61-162 ~]# mysql --no-defaults --help|egrep "user|host|port"
-h, --host=name        Connect to host.
-P, --port=#           Port number to use for connection or 0 for default to, in
-u, --user=name        User for login if not current user.
host                   127.0.0.1
port                   3306
user                   zhenxing

## 查看读取的所有客户端配置文件参数设置
[root@10-186-61-162 ~]# mysql --print-defaults
mysql would have been started with the following arguments:
--host=127.0.0.1 --user=root --port=3306 --host=127.0.0.1 --user=root --port=3306 --prompt=\U[\d]
```

从上面输出的结果来看,我们可以得到以下2个基本现象:

- 即使指定 `--no-defaults` 不读取任何配置文件,这个 `user` 的默认值依旧是 `zhenxing` 用户
- 当输出 `--print-defaults` 获取实际运行值时, 可以看到 `/etc/my.cnf` 下的 `[client]`, 和 `[mysql]` 标签下的属性配置从上到下被正确获取
- 除了 `/etc/my.cnf` 外, 在最后还有 `--user=zhenxing --password=***** --host=127.0.0.1 --port=3306` 这4个参数被额外添加到了命令最后

**ps:** mysql 客户端和服务端读取配置的原则都是 文件读取从上到下,后面相同参数配置覆盖前面的参数

经过一系列的排除, 依旧没找到这个默认值被修改的源头

## 8、打印 mysql 客户端的系统调用

- 使用 `strace` 直接观测 mysql 客户端在执行时到底调用了哪些配置, 以下是调用 `my.cnf` 相关配置的片段 (对结果做了精简输出)

```
1. stat("/etc/my.cnf", {st_mode=S_IFREG|0644, st_size=195, ...}) = 0      = 3
2. stat("/etc/mysql/my.cnf", 0x7ffd56813180) = -1 ENOENT (No such file or directory)
3. stat("/usr/local/mysql/etc/my.cnf", 0x7ffd56813180) = -1 ENOENT (No such file or directory)
4. stat("/data/mysql/3306/base/my.cnf", 0x7ffd56813180) = -1 ENOENT (No such file or directory)
5. stat("/root/.my.cnf", 0x7ffd56813180) = -1 ENOENT (No such file or directory)
6. stat("/root/.mylogin.cnf", {st_mode=S_IFREG|0600, st_size=336, ...}) = 0
```

通过以上调用顺序可以可到以下结论:

- 1-5行的调用顺序与我们验证的逻辑基本一致
- 2-5行显示为 `No such file or directory` 与我们的验证结果一致
- 第6行输出, 增加了一个对 `/root/.mylogin.cnf` 的读取操作, 并且可以知道当前这个文件是确实存在的

## 9、排查 `/root/.mylogin.cnf`

看到这个文件我们一般都知道, 这个是 `mysql_config_editor` 工具用来配置 `login-path` 的生成的文件, 我们可以用以下方式查看当前的配置信息

```
[root@10-186-61-162 ~]# mysql_config_editor print --all
[client]
user = "zhenxing"
password = *****
host = "127.0.0.1"
port = 3306
```

这里可以看到配置中有一个 `client` 标签的连接参数配置,配置的内容正好是我们文章开头显示的异常默认值,到这我们基本定位了造成这个奇怪现象的原因,出现这个故障场景的原因也是刚好这台是测试环境曾经做过一些 `mysql_config_editor` 用法的测试,导致了该现象的发生

### 三、场景总结

1、mysql客户端除了会按照命令 `mysql --verbose --help|grep my.cnf` 输出的常规的顺序读取配置外,在最后还会额外的读取 `.mylogin.cnf` 文件中配置

2、即使指定了 `--no-defaults`, 依旧会去读取 `.mylogin.cnf` 中的 `[client]`, `[mysql]` 标签的配置值

其中官方文档也在以下链接中给到了明确的说明(以下是关键描述片段)

- <https://dev.mysql.com/doc/refman/8.0/en/option-file-options.html>

The mysql client reads `[client]` and `[mysql]` from other option files, and `[client]`, `[mysql]`, and `[mysqlx]`

Client programs **read** the login path file even when the `--no-defaults` option is used. This permits

- 个人猜测当时这么设计的目的是考虑如备份脚本等需要连接数据库时,为了防止非必要的参数文件的干扰,在指定了 `--no-defaults` 参数后依旧能使用到 `.mylogin.cnf` 文件中配置的加密密码,提升安全性。

---

#### 文章推荐:

MySQL 设置 `terminology_use_previous` 参数导致数据库 Crash

MySQL 主从复制中创建复制用户的时机探讨

分布式 | Prepare Statement 协议游标可行性

---





本文关键字：#mysql客户端# #参数默认值 login-path#

💖 点一下“阅读原文”了解更多资讯

阅读原文

喜欢此内容的人还喜欢

解决ASP.NET Core在Task中使用IServiceProvider的问题  
yi念之间



太全了！用Python操作MySQL的使用教程集锦！  
杰哥的IT之旅



MySQL性能调优，必须掌握这一个工具！！！（1分钟系列）  
架构师之路

