Rocky Enterprise Linux 9.2 Manual Pages on command 'mysql_secure_installation.1'

$ man mysql_secure_installation.1

MYSQL_SECURE_INSTALLATION(1)   MySQL Database System   MYSQL_SECURE_INSTALLATION(1)

NAME

   mysql_secure_installation - improve MySQL installation security

SYNOPSIS

   mysql_secure_installation

DESCRIPTION

   This program enables you to improve the security of your MySQL installation in the

   following ways:

   ?   You can set a password for root accounts.

   ?   You can remove root accounts that are accessible from outside the local host.

   ?   You can remove anonymous-user accounts.

   ?   You can remove the test database (which by default can be accessed by all users, even

       anonymous users), and privileges that permit anyone to access databases with names

       that start with test_.

   mysql_secure_installation helps you implement security recommendations similar to those

   described at Section 2.9.4, ?Securing the Initial MySQL Account?.

   Normal usage is to connect to the local MySQL server; invoke mysql_secure_installation

   without arguments:

       mysql_secure_installation

   When executed, mysql_secure_installation prompts you to determine which actions to

   perform.

   The validate_password component can be used for password strength checking. If the plugin

   is not installed, mysql_secure_installation prompts the user whether to install it. Any

passwords entered later are checked using the plugin if it is enabled.

Most of the usual MySQL client options such as --host and --port can be used on the command line and in option files. For example, to connect to the local server over IPv6 using port 3307, use this command:

    mysql_secure_installation --host=::1 --port=3307

mysql_secure_installation supports the following options, which can be specified on the command line or in the [mysql_secure_installation] and [client] groups of an option file. For information about option files used by MySQL programs, see Section 4.2.2.2, ?Using Option Files?.

? --help, -?

????????????????????????????????

?Command-Line Format ? --help ?

???????????????????????????????

Display a help message and exit.

? --defaults-extra-file=file_name

??

?Command-Line Format ? --defaults-extra-file=file_name ?

???

?Type   ? File name   ?

??

Read this option file after the global option file but (on Unix) before the user

option file. If the file does not exist or is otherwise inaccessible, an error occurs.

If file_name is not an absolute path name, it is interpreted relative to the current

directory.

For additional information about this and other option-file options, see

Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --defaults-file=file_name

??

?Command-Line Format ? --defaults-file=file_name ?

???

?Type   ? File name   ?

??

Use only the given option file. If the file does not exist or is otherwise

inaccessible, an error occurs. If file_name is not an absolute path name, it is interpreted relative to the current directory.

For additional information about this and other option-file options, see Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --defaults-group-suffix=str

??

?Command-Line Format ? --defaults-group-suffix=str ?

??

?Type   ? String   ?

??

Read not only the usual option groups, but also groups with the usual names and a suffix of str. For example, mysql_secure_installation normally reads the [client] and [mysql_secure_installation] groups. If this option is given as --defaults-group-suffix=_other, mysql_secure_installation also reads the [client_other] and [mysql_secure_installation_other] groups.

For additional information about this and other option-file options, see Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --host=host_name, -h host_name

??????????????????????????????????????

?Command-Line Format ? --host ?

??????????????????????????????????????

Connect to the MySQL server on the given host.

? --no-defaults

??

?Command-Line Format ? --no-defaults ?

??

Do not read any option files. If program startup fails due to reading unknown options from an option file, --no-defaults can be used to prevent them from being read. The exception is that the .mylogin.cnf file is read in all cases, if it exists. This permits passwords to be specified in a safer way than on the command line even when --no-defaults is used. To create .mylogin.cnf, use the mysql_config_editor utility. See mysql_config_editor(1).

For additional information about this and other option-file options, see

Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

?   --password=password, -p password

???

?Command-Line Format ? --password=password ?

???

?Type   ? String   ?

???

?Default Value   ? [none]   ?

???

This option is accepted but ignored. Whether or not this option is used,

mysql_secure_installation always prompts the user for a password.

?   --port=port_num, -P port_num

???

?Command-Line Format ? --port=port_num ?

???

?Type   ? Numeric   ?

???

?Default Value   ? 3306   ?

???

For TCP/IP connections, the port number to use.

?   --print-defaults

???

?Command-Line Format ? --print-defaults ?

???

Print the program name and all options that it gets from option files.

For additional information about this and other option-file options, see

Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

?   --protocol={TCP|SOCKET|PIPE|MEMORY}

???

?Command-Line Format ? --protocol=type   ?

???

?Type   ? String   ?

???

Default Value	[see text]
Valid Values	
	TCP
	SOCKET
	PIPE
	MEMORY

The transport protocol to use for connecting to the server. It is useful when the other connection parameters normally result in use of a protocol other than the one you want. For details on the permissible values, see Section 4.2.7, ?Connection Transport Protocols?.

? --socket=path, -S path

Command-Line Format	--socket={file_name|pipe_name}
Type	String

For connections to localhost, the Unix socket file to use, or, on Windows, the name of the named pipe to use.

On Windows, this option applies only if the server was started with the named_pipe system variable enabled to support named-pipe connections. In addition, the connection must be a member of the Windows group specified by the named_pipe_full_access_group system variable.

? --ssl* Options that begin with --ssl specify whether to connect to the server using encryption and indicate where to find SSL keys and certificates. See the section called ?Command Options for Encrypted Connections?.

? --ssl-fips-mode={OFF|ON|STRICT}

Command-Line Format	--ssl-fips-mode={OFF|ON|STRICT}

Deprecated	8.0.34	
Type	Enumeration	
Default Value	OFF	
Valid Values		
		OFF
		ON
		STRICT

Controls whether to enable FIPS mode on the client side. The --ssl-fips-mode option differs from other --ssl-xxx options in that it is not used to establish encrypted connections, but rather to affect which cryptographic operations to permit. See Section 6.8, ?FIPS Support?.

These --ssl-fips-mode values are permitted:

? OFF: Disable FIPS mode.

? ON: Enable FIPS mode.

? STRICT: Enable ?strict? FIPS mode.

  Note

  If the OpenSSL FIPS Object Module is not available, the only permitted value for --ssl-fips-mode is OFF. In this case, setting --ssl-fips-mode to ON or STRICT causes the client to produce a warning at startup and to operate in non-FIPS mode.

As of MySQL 8.0.34, this option is deprecated. Expect it to be removed in a future version of MySQL.

? --tls-ciphersuites=ciphersuite_list

Command-Line Format	--tls-ciphersuites=ciphersuite_list

?Introduced   ? 8.0.16   ?

??

?Type   ? String   ?

???

The permissible ciphersuites for encrypted connections that use TLSv1.3. The value is

a list of one or more colon-separated ciphersuite names. The ciphersuites that can be

named for this option depend on the SSL library used to compile MySQL. For details,

see Section 6.3.2, ?Encrypted Connection TLS Protocols and Ciphers?.

This option was added in MySQL 8.0.16.

?   --tls-version=protocol_list

??

?Command-Line Format   ? --tls-version=protocol_list   ?

??

?Type   ? String   ?

??

?Default Value (? 8.0.16) ?   ?

?   ?   TLSv1,TLSv1.1,TLSv1.2,TLSv1.3 ?

?   ?   (OpenSSL 1.1.1 or   ?

?   ?   higher)   ?

?   ?   ?

?   ?   TLSv1,TLSv1.1,TLSv1.2   ?

?   ?   (otherwise)   ?

??

?Default Value (? 8.0.15) ? TLSv1,TLSv1.1,TLSv1.2   ?

??

The permissible TLS protocols for encrypted connections. The value is a list of one or

more comma-separated protocol names. The protocols that can be named for this option

depend on the SSL library used to compile MySQL. For details, see Section 6.3.2,

?Encrypted Connection TLS Protocols and Ciphers?.

?   --use-default

???

?Command-Line Format ? --use-default ?

???

?Type   ? Boolean   ?

???

Execute noninteractively. This option can be used for unattended installation

operations.

?   --user=user_name, -u user_name

??

?Command-Line Format ? --user=user_name ?

??

?Type   ? String   ?

??

The user name of the MySQL account to use for connecting to the server.

COPYRIGHT

SEE ALSO

For more information, please refer to the MySQL Reference Manual, which may already be

installed locally and which is also available online at http://dev.mysql.com/doc/.

AUTHOR

Oracle Corporation (http://dev.mysql.com/).

MySQL 8.0   11/27/2023   MYSQL_SECURE_INSTALLATION(1)