

[NEWS] General Security Guidelines (MySQL and SQL Web Interfaces)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2001-09/0091.html>

From: support@securiteam.com

Date: 09/29/01

From: support@securiteam.com

To: list@securiteam.com

Subject: [NEWS] General Security Guidelines (MySQL and SQL Web Interfaces)

Message-Id: <20010929215108.F1DA9138BF@mail.der-keiler.de>

Date: Sat, 29 Sep 2001 23:51:08 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> – Know that you're safe.

General Security Guidelines (MySQL and SQL Web Interfaces)

SUMMARY

Anyone using MySQL (or any other SQL server) on a computer connected to the Internet should read this advisory to avoid the most common security mistakes.

The following is MySQL specific; however it can be used to test your specific SQL implementation.

DETAILS

In discussing security, it is important to emphasize the necessity of fully protecting the entire server host (not simply the MySQL server) against all types of applicable attacks: eavesdropping, altering, playback, and denial of service. We do not cover all aspects of availability and fault tolerance here.

MySQL uses security based on Access Control Lists (ACLs) for all connections, queries, and other operations that a user may attempt to perform. There is also some support for SSL-encrypted connections between

MySQL clients and servers. Many of the concepts discussed here are not specific to MySQL at all; the same general ideas apply to almost all applications.

When running MySQL, follow these guidelines whenever possible:

- * don't ever give anyone (except the MySQL root user) access to the USER table in the MySQL database! The encrypted password is the real password in MySQL. If you know the password listed in the user table for a given user, you can easily log in as that user if you have access to the host listed for that account.

- * Learn the MySQL access privilege system. The GRANT and REVOKE commands are used for controlling access to MySQL. Do not grant any more privileges than necessary. Never grant privileges to all hosts. Checklist:

- * Try `mysql -u root`. If you are able to connect successfully to the server without being asked for a password, you have problems. Anyone can connect to your MySQL server as the MySQL root user with full privileges! Review the MySQL installation instructions, paying particular attention to the item about setting a root password.

- * Use the command `SHOW GRANTS` and check to see who has access to what. Remove those privileges that are not necessary using the `REVOKE` command.

- * Do not keep any plain-text passwords in your database. When your computer becomes compromised, the intruder can take the full list of passwords and use them. Instead use MD5() or another one-way hashing function.

- * Do not choose passwords from dictionaries. There are special programs to break them. Even passwords like ```xfish98`" are very bad. Much better is ```duag98`" which contains the same word ```fish`" but typed one key to the left on a standard QWERTY keyboard. Another method is to use ```Mhall`" which is taken from the first characters of each word in the sentence ```Mary had a little lamb.`" This is easy to remember and type, but difficult to guess for someone who does not know it.

- * Invest in a firewall. This protects you from at least 50% of all types of exploits in any software. Put MySQL behind the firewall or in a demilitarized zone (DMZ). Checklist:

- * Try to scan your ports from the Internet using a tool such as <http://www.securiteam.com/tools/3Z5QNPPNFA.html> NMap. MySQL uses port 3306 by default. This port should be inaccessible from untrusted hosts.

Another simple way to check whether your MySQL port is open is to try the following command from some remote machine, where `server_host` is the hostname of your MySQL server:

```
shell> telnet server_host 3306
```

If you get a connection and some garbage characters, the port is open, and should be closed on your firewall or router, unless you really have a good reason to keep it open. If telnet just hangs or the connection is refused, everything is OK; the port is blocked.

* Do not trust any data entered by your users. They can try to trick your code by entering special or escaped character sequences in Web forms, URLs, or whatever application you have built. Be sure that your application remains secure if a user enters something like ``; DROP DATABASE mysql;". This is an extreme example, but large security leaks and data loss may occur because of hackers using similar techniques, if you do not prepare for them. Also, remember to check numeric data. A common mistake is to protect only strings. Sometimes people think that if a database contains only publicly available data that it need not be protected. This is incorrect. At least denial-of-service type attacks can be performed on such databases. The simplest way to protect from this type of attack is to use apostrophes around the numeric constants: SELECT * FROM table WHERE ID='234' rather than SELECT * FROM table WHERE ID=234. MySQL automatically converts this string to a number and strips all non-numeric symbols from it.

Checklist:

* All Web applications:

* Try to enter ``" and ``"' in all your Web forms. If you get any kind of MySQL error, investigate the problem right away.

* Try to modify any dynamic URLs by adding %22 (^"), %23 (^#), and %27 (^') in the URL.

* Try to modify data types in dynamic URLs from numeric ones to character ones containing characters from previous examples. Your application should be safe against this and similar attacks.

* Try to enter characters, spaces, and special symbols instead of numbers in numeric fields. Your application should remove them before passing them to MySQL or your application should generate an error. Passing unchecked values to MySQL is very dangerous!

* Check data sizes before passing them to MySQL.

* Consider having your application connect to the database using a different user name than the one you use for administrative purposes. Do not give your applications any more access privileges than they need.

* Users of PHP:

* Check out the addslashes() function. As of PHP 4.0.3, a mysql_escape_string() function is available that is based on the function of the same name in the MySQL C API.

* Users of MySQL C API:

* Check out the mysql_escape_string() API call.

* Users of MySQL++:

* Check out the escape and quote modifiers for query streams.

* Users of Perl DBI:

* Check out the quote() method or use placeholders.

* Users of Java JDBC:

* Use a PreparedStatement object and placeholders.

Securiteam: [NEWS] General Security Guidelines (MySQL and SQL W

* Do not transmit plain (unencrypted) data over the Internet. These data are accessible to everyone who has the time and ability to intercept it and use it for their own purposes. Instead, use an encrypted protocol such as SSL or SSH. MySQL supports internal SSL connections as of Version 3.23.9. SSH port-forwarding can be used to create an encrypted (and compressed) tunnel for the communication.

* Learn to use the tcpdump and strings utilities. For most cases, you can check whether or not MySQL data streams are unencrypted by issuing a command like the following:
shell> tcpdump -l -i eth0 -w - src or dst port 3306 | strings

(This works under Linux and should work with small modifications under other systems). Warning: If you do not see data this does not always actually mean that, it is encrypted. If you need high security, you should consult with a security expert.

ADDITIONAL INFORMATION

The information has been provided by
<http://www.mysql.com/doc/G/e/General_security.html> www.MySQL.com.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- **Previous message:** support@securiteam.com: "[NEWS] Cisco Secure PIX Firewall SMTP Filtering Vulnerability (Regression Problem)"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)