Advanced Information Security

Corporation

10/07/2015

Advanced Information Security Corporation

Security Advisory Report

MySQL Database 5.6.x (LATEST) Security Report

Software Security Notification

Software: **Oracle's MySQL v.5.6.24 (LATEST)**

Vulnerability:

(11) Buffer Overflow Vulnerabilities / ~ Deprecated & Insecure Function use (Missing Bounds-checks)

Software Overview

MySQL is an open source relational database management system (RDBMS), and the world's most popular Open Source database. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack. MySQL is used in many high profile large-websites worldwide. MySQL was created by a Swedish company, MySQL AB founded by David Axmark, Allan Larsson and Michael "Monty" Widenius. The first version of the software appeared on 23 of May, 1995. Oracle Corporation acquired Sun Microsystems in April, 2009 and are now the owners of MySQL Copyright and Trademark.

Summary

During a manual source-code audit of Oracle's MySQL v 5.6.24 database; conducted internally by the Advanced Information Security Group, instances of insecure function calls were observed in the software. The issues stem from the lack of any manual control metrics, which would prevent data from being overwritten into other sensitive locations.

Code Snippet –- [/usr/bin/mysql_plugin] -–
(../mysql/mysql-5.6.24/client/mysql_plugin.c:796)

```
793     /* read the plugin config file and check for match against argument */
794     else
795     {
796       strcpy(plugin_name, argv[i]);
797       strcpy(config_file, argv[i]);
798       strcat(config_file, ".ini");
799     }
800   }
801
```

Description

Unsafe use of the **strcpy()** function, has been triggered resulting in a buffer overflow condition. Therefore, in the aforementioned experiment input is copied from the command-line, to a fixed length destination buffer. No bounds checks are provided to ensure that the source does not exceed in size, and therefore would not overwrite the destination buffer.

Technical Details

```
Applications  Places  Terminal  Wed 19:48
root@kali: ~
File  Edit  View  Search  Terminal  Help
*** buffer overflow detected ***: /usr/bin/mysql_plugin terminated
======= Backtrace: =========
/lib/i386-linux-gnu/i686/cmov/libc.so.6(+0x6c6f3)[0xb7ce86f3]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__fortify_fail+0x45)[0xb7d762d5]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(+0xf838a)[0xb7d7438a]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__strcpy_chk+0x37)[0xb7d73877]
/usr/bin/mysql_plugin(main+0x202)[0x80009e22]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__libc_start_main+0xf3)[0xb7c95a63]
/usr/bin/mysql_plugin(+0xa90d)[0x8000a90d]
======= Memory map: ========
80000000-8023c000 r-xp 00000000 00:13 102356     /usr/bin/mysql_plugin
8023c000-8023f000 r--p 0023b000 00:13 102356     /usr/bin/mysql_plugin
8023f000-802b5000 rw-p 0023e000 00:13 102356     /usr/bin/mysql_plugin
802b5000-802d9000 rw-p 00000000 00:00 0          [heap]
b7300000-b7321000 rw-p 00000000 00:00 0
b7321000-b7400000 ---p 00000000 00:00 0
b7478000-b7479000 ---p 00000000 00:00 0
b7479000-b7c7c000 rw-p 00000000 00:00 0
b7c7c000-b7e20000 r-xp 00000000 00:13 1673       /lib/i386-linux-gnu/i686/cmov/libc-2.1
9.so
b7e20000-b7e22000 r--p 001a4000 00:13 1673       /lib/i386-linux-gnu/i686/cmov/libc-2.1
9.so
b7e22000-b7e23000 rw-p 001a6000 00:13 1673       /lib/i386-linux-gnu/i686/cmov/libc-2.1
9.so
b7e23000-b7e26000 rw-p 00000000 00:00 0
b7e26000-b7e42000 r-xp 00000000 00:13 15850      /lib/i386-linux-gnu/libgcc_s.so.1
b7e42000-b7e43000 rw-p 0001b000 00:13 15850      /lib/i386-linux-gnu/libgcc_s.so.1
b7e43000-b7e87000 r-xp 00000000 00:13 15645      /lib/i386-linux-gnu/i686/cmov/libm-2.1
```

Technical Details

```
Applications  Places  Terminal  Fri 19:18
root@kali: ~
File  Edit  View  Search  Terminal  Help
b7fb8000-b7fba000 rw-p 00000000 00:00 0
b7fd9000-b7fdc000 rw-p 00000000 00:00 0
b7fdc000-b7fde000 r--p 00000000 00:00 0          [vvar]
b7fde000-b7fdf000 r-xp 00000000 00:00 0          [vdso]
b7fdf000-b7ffe000 r-xp 00000000 00:13 1667       /lib/i386-linux-gnu/ld-2.19.so
b7ffe000-b7fff000 r--p 0001f000 00:13 1667       /lib/i386-linux-gnu/ld-2.19.so
b7fff000-b8000000 rw-p 00020000 00:13 1667       /lib/i386-linux-gnu/ld-2.19.so
bffc9000-c0000000 rw-p 00000000 00:00 0          [stack]

Program received signal SIGABRT, Aborted.
0xb7fdebe0 in __kernel_vsyscall ()
(gdb) bt
#0  0xb7fdebe0 in __kernel_vsyscall ()
#1  0xb7caa307 in __GI_raise (sig=sig@entry=6)
    at ../nptl/sysdeps/unix/sysv/linux/raise.c:56
#2  0xb7cab9c3 in __GI_abort () at abort.c:89
#3  0xb7ce86f8 in __libc_message (do_abort=do_abort@entry=2,
    fmt=fmt@entry=0xb7ddbe55 "*** %s ***: %s terminated\n")
    at ../sysdeps/posix/libc_fatal.c:175
#4  0xb7d762d5 in __GI___fortify_fail (
    msg=msg@entry=0xb7ddbdd6 "buffer overflow detected")
    at fortify_fail.c:31
#5  0xb7d7438a in __GI___chk_fail () at chk_fail.c:28
#6  0xb7d73877 in __strcpy_chk (dest=0xbffe8c9c 'A' <repeats 200 times>...,
    src=0xbffe96ed 'A' <repeats 200 times>..., destlen=<optimized out>)
    at strcpy_chk.c:60
#7  0x80009e22 in main ()
(gdb)
```

Technical Details

```
(gdb) disas
Dump of assembler code for function __kernel_vsyscall:
   0xb7fdebd0 <+0>:     push   %ecx
   0xb7fdebd1 <+1>:     push   %edx
   0xb7fdebd2 <+2>:     push   %ebp
   0xb7fdebd3 <+3>:     mov    %esp,%ebp
   0xb7fdebd5 <+5>:     sysenter
   0xb7fdebd7 <+7>:     nop
   0xb7fdebd8 <+8>:     nop
   0xb7fdebd9 <+9>:     nop
   0xb7fdebda <+10>:    nop
   0xb7fdebdb <+11>:    nop
   0xb7fdebdc <+12>:    nop
   0xb7fdebdd <+13>:    nop
   0xb7fdebde <+14>:    int    $0x80
=> 0xb7fdebe0 <+16>:    pop    %ebp
   0xb7fdebe1 <+17>:    pop    %edx
   0xb7fdebe2 <+18>:    pop    %ecx
   0xb7fdebe3 <+19>:    ret
End of assembler dump.
(gdb)
```

Proof of Concept Exploit – MySQL v5.6.24

```
Applications ▾  Places ▾  Terminal ▾  Wed 21:54
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# /usr/bin/mysql_plugin `perl -e 'print "A" x 9000'`
*** buffer overflow detected ***: /usr/bin/mysql_plugin terminated
======= Backtrace: =========
/lib/i386-linux-gnu/i686/cmov/libc.so.6(+0x6c6f3)[0xb71e46f3]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__fortify_fail+0x45)[0xb72722d5]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(+0xf838a)[0xb727038a]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__strcpy_chk+0x37)[0xb726f877]
/usr/bin/mysql_plugin(main+0x202)[0xb7505e22]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__libc_start_main+0xf3)[0xb7191a63]
/usr/bin/mysql_plugin(+0xa90d)[0xb750690d]
======= Memory map: ========
b6800000-b6821000 rw-p 00000000 00:00 0
b6821000-b6900000 ---p 00000000 00:00 0
b6974000-b6975000 ---p 00000000 00:00 0
b6975000-b7178000 rw-p 00000000 00:00 0
b7178000-b731c000 r-xp 00000000 00:13 1673       /lib/i386-linux-gnu/i686/cmov/libc-2.1
9.so
b731c000-b731e000 r--p 001a4000 00:13 1673       /lib/i386-linux-gnu/i686/cmov/libc-2.1
9.so
b731e000-b731f000 rw-p 001a6000 00:13 1673       /lib/i386-linux-gnu/i686/cmov/libc-2.1
9.so
b731f000-b7322000 rw-p 00000000 00:00 0
b7322000-b733e000 r-xp 00000000 00:13 15851      /lib/i386-linux-gnu/libgcc_s.so.1
b733e000-b733f000 rw-p 0001b000 00:13 15851      /lib/i386-linux-gnu/libgcc_s.so.1
b733f000-b7383000 r-xp 00000000 00:13 15639      /lib/i386-linux-gnu/i686/cmov/libm-2.1
9.so
b7383000-b7384000 r--p 00043000 00:13 15639      /lib/i386-linux-gnu/i686/cmov/libm-2.1
9.so
```

Technical Synopsis

The cause of the issue is an unsafe strcpy call; which can lead to a buffer overflow condition. A user-supplied string from the command-line is copied to a fixed length destination buffer, and the vulnerable function call is demonstrated below in red.

```
======= Backtrace: =========
/lib/i386-linux-gnu/i686/cmov/libc.so.6(+0x6c6f3)[0xb71e46f3]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__fortify_fail+0x45)[0xb72722d5]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(+0xf838a)[0xb727038a]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__strcpy_chk+0x37)[0xb726f877]
/usr/bin/mysql_plugin(main+0x202)[0xb7505e22]
/lib/i386-linux-gnu/i686/cmov/libc.so.6(__libc_start_main+0xf3)[0xb7191a63]
/usr/bin/mysql_plugin(+0xa90d)[0xb750690d]
```

Source Code at Line: 796
File: (../mysql/mysql-5.6.24/client/mysql_plugin.c)

MYSQL v5.6.24 Vulnerability List

1. Main.c (../mysql/mysql-5.6.24/regex/main.c:577)

```
572  char *name;
573  {
574          static char efbuf[100];
575          my_regex_t re;
576
577          sprintf(efbuf, "MY_REG %s", name);
578          assert(strlen(efbuf) < sizeof(efbuf));
579          re.re_endp = efbuf;
580          (void) my_regerror(MY_REG_ATOI, &re, efbuf, sizeof(efbuf));
581          return(atoi(efbuf));
582  }
```

Description: A char* type is copied to a fixed length destination buffer. This could lead to a buffer overflow.

2. Code Snippet – mysql_plugin.c (../mysql/mysql-5.6.24/client/mysql_plugin.c:796)

```
793     /* read the plugin config file and check for match against argument */
794     else
795     {
796       strcpy(plugin_name, argv[i]);
797       strcpy(config_file, argv[i]);
798       strcat(config_file, ".ini");
799     }
800   }
801
```

Description: Unsafe use of strcpy could lead to an overflow condition. A user-supplied string from the command-line is copied to a fixed length destination buffer.

3. Code Snippet – mysql_plugin.c (../mysql/mysql-5.6.24/client/mysql_plugin.c:797)

```
793     /* read the plugin config file and check for match against argument */
794     else
795     {
796       strcpy(plugin_name, argv[i]);
797       strcpy(config_file, argv[i]);
798       strcat(config_file, ".ini");
799     }
800   }
801
```

Description: Unsafe Use of strcpy could lead to an overflow condition. A user-supplied string from the command-line is copied to a fixed length destination buffer. This could lead to a buffer overflow.

4. Code Snippet – main.c (../mysql/mysql-5.6.24/regex/main.c:544)

```
542                 shlen = 1;      /* force check for end-of-string */
543         if (strncmp(p, at, shlen) != 0) {
544                 sprintf(grump, "matched null at `%.20s'", p);
545                 return(grump);
546         }
547         return(NULL);
548 }
```

Description: A char* type is being copied to a fixed length destination buffer. This could lead to a buffer overflow.

5. Code Snippet – main.c (../mysql/mysql-5.6.24/regex/main.c:525)

```
519         len = (int)(sub.rm_eo - sub.rm_so);
520         shlen = (int)strlen(should);
521         p = str + sub.rm_so;
522
523         /* check for not supposed to match */
524         if (should == NULL) {
525                 sprintf(grump, "matched `%.*s'", len, p);
526                 return(grump);
527         }
```

Description: Insecure sprintf. A char* type is being copied to a fixed length destination buffer. This could lead to a buffer overflow.

6. Code Snippet – reader.cpp (../mysql/mysql-5.6.24/storage/ndb/src/kernel/blocks/dblqh/redoLogReader/reader.cpp:413)

```
406 void readArguments(int argc, const char** argv)
407 {
408   if(argc < 2 || argc > 9){
409     usage(argv[0]);
410     doExit();
411   }
412
413   strcpy(fileName, argv[1]);
```

Description: Unsafe use of strcpy could lead to an overflow condition. A user-supplied string from the command-line is written to a fixed length destination buffer. This could lead to a buffer overflow if the input provided, is of greater size than the destination buffer.

7. Code Snippet – main.c (../mysql/mysql-5.6.24/regex/main.c:531)

```
528
529         /* check for wrong match */
530         if (len != shlen || strncmp(p, should, (size_t)shlen) != 0) {
531                 sprintf(grump, "matched `%.*s' instead", len, p);
532                 return(grump);
533         }
```

Description: Unsafe use of strcpy could lead to an overflow condition. A char* type is being copied to a fixed length destination buffer. This could lead to a buffer overflow.

8. Code Snippet – mysqlshow.c (../mysql/mysql-5.6.24/client/mysqlshow.c:710)

```
701   if (mysql_select_db(mysql,db))
702   {
703     fprintf(stderr,"%s: Cannot connect to db: %s: %s\n",my_progname,db,
704             mysql_error(mysql));
705     return 1;
706   }
707
708   if (opt_count)
709   {
710     sprintf(query,"select count(*) from `%s`", table);
711     if (mysql_query(mysql,query) || !(result=mysql_store_result(mysql)))
712     {
713       fprintf(stderr,"%s: Cannot get record count for db: %s, table: %s: %s\n",
714               my_progname,db,table,mysql_error(mysql));
715       return 1;
716     }
717     row= mysql_fetch_row(result);
718     rows= (ulong) strtoull(row[0], (char**) 0, 10);
719     mysql_free_result(result);
720   }
```

Description: Insecure use of sprintf. A char* type is being copied to a fixed length destination buffer. This could lead to a buffer overflow.

9. Code Snippet – conf_to_src.c (../mysql/mysql-5.6.24/libmysql/conf_to_src.c:121

```
116  void
117  print_arrays_for(char *set)
118  {
119    FILE *f;
120
121    sprintf(buf, "%s.conf", set);
122
123    if ((f = fopen(buf, "r")) == NULL) {
124      fprintf(stderr, "%s: can't read conf file for charset %s\n", prog, set);
125      exit(EXIT_FAILURE);
126    }
127
```

Description: Insecure use of sprintf. A char* type is being copied to a fixed length destination buffer. This could lead to a buffer overflow.

10. Code Snippet – PosixAsyncFile.ccp (../mysql/mysql-5.6.24/storage/ndb/src/kernel/blocks/ndbfs/PosixAsyncFile.cpp:784)

```
772  void
773  PosixAsyncFile::rmrfReq(Request *request, const char * src, bool removePath)
774  {
775    if(!request->par.rmrf.directory)
776    {
777      // Remove file
778      if(unlink(src) != 0 && errno != ENOENT)
779        request->error = errno;
780      return;
781    }
782
783    char path[PATH_MAX];
784    strcpy(path, src);
785    strcat(path, "/");
```

Description: Unsafe Use of strcpy could lead to an overflow condition. A char* type is being copied to a fixed length destination buffer. This could lead to a buffer overflow.

11. Code Snippet – Win32AsyncFile.ccp (../mysql/mysql-5.6.24/storage/ndb/src/kernel/blocks/ndbfs/Win32AsyncFile.cpp:377)

```
362  void
363  Win32AsyncFile::rmrfReq(Request * request, const char * src, bool removePath){
364    if (!request->par.rmrf.directory)
365    {
366      // Remove file
367      if (!DeleteFile(src))
368      {
369        DWORD dwError = GetLastError();
370        if (dwError != ERROR_FILE_NOT_FOUND)
371          request->error = dwError;
372      }
373      return;
374    }
375
376    char path[PATH_MAX];
377    strcpy(path, src);
378    strcat(path, "\\*");
379
380    WIN32_FIND_DATA ffd;
381    HANDLE hFindFile;
```

<<<

Size of PATH is PATH_MAX 256

Description: Unsafe Use of strcpy could lead to an overflow condition. A char* type is being copied to a fixed length destination buffer. This, could lead to an overflow.

Acknowledgements

Sincere Thanks to Oracle Corporation for their excellent cooperation.

References

[1]Security Focus Website (2015). *Advanced Information Security Corporation, Security Advisory - MySQL v5.6.24 Buffer Overflows*. [Online] Available at: http://www.securityfocus.com/archive/1/536634 [Accessed 7 Oct. 2015].